

Rijden onder invloed



Bijlage 1: Zelf een computerspel maken

Zelf een computerspel maken

Inleiding

Het spelen van computerspellen is leuk. Maar wat eigenlijk veel leuker is, is het maken van je eigen computerspellen en anderen die laten spelen. Helaas is het maken van computerspellen niet eenvoudig. Dankzij het softwarepakket Gamemaker kun je al snel heel aantrekkelijke tweedimensionale computerspellen in elkaar steken. Op die manier leer je spelenderwijs de technieken van het programmeren.

Doelstelling

Deze module wil je begeleiden in het zelf leren ontwikkelen van een computerspel.

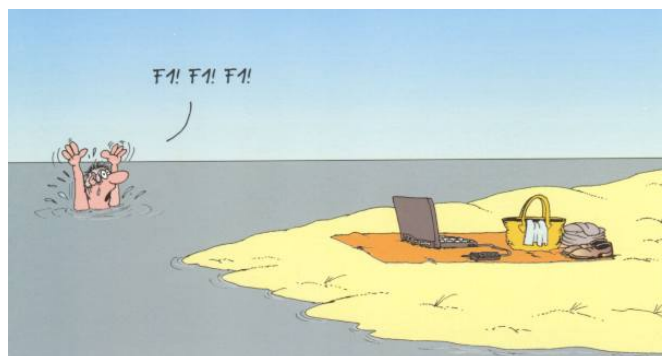
Het uiteindelijke doel is een computerspel ontwikkelen waarbij de moeilijkheidsgraad verandert naarmate de speler meer alcohol heeft gedronken.

Na het doornemen van deze module zal je in staat zijn om

- je basiskennis van het programmeren in TI-Basic te vertalen naar de ontwikkeling van een computerspel: werken met variabelen, opstellen van IF-THEN-ELSE-testen ...;
- zelf tweedimensionale computerspellen ontwikkelen

Studiemateriaal

De leerstof voor deze module wordt je aangeboden in de vorm van een werktekst. De tekst start de installatie van de nodige software die je kunt downloaden van het net. Na een eerste kennismaking wordt onmiddellijk gestart met het ontwikkelen van een eenvoudig computerspel. Nadien komt een tweede meer ingewikkeld spel aan bod. Na deze twee voorbeelden ben je in staat om je eigen spel te bedenken en te ontwikkelen.



Deze werktekst is gebaseerd op de teksten die de website www.gamemaker.nl ter beschikking stelt. Deze handleiding beperkt zich tot de meest noodzakelijke informatie om snel tot een resultaat te komen en de beginselen van het programmeren op een duidelijke wijze te ervaren. Je kunt best de volledige documentatie bij de hand hebben om bij vragen die je jezelf stelt wat meer uitgebreide informatie te lezen. De volledige handleiding kun je downloaden van de website www.gamemaker.nl uit de rubriek "documentation".

Het programma Gamemaker 6 installeren

Om *Game Maker 6* te kunnen gebruiken heb je een moderne Pentium PC nodig met minimaal Windows98 of hoger. Het vereist een beeldschermresolutie van minimaal 800x600 en 65000(16-bit) kleuren. Bovendien moet DirectX op je PC zijn geïnstalleerd. Om spellen te maken en te testen moet er minimaal 32MB RAM aanwezig zijn, liefst meer. Om alleen maar spelletjes te spelen kan ook minder RAM-geheugen volstaan. Wel hangt het van het type spel af hoeveel geheugen nodig blijkt te zijn. Heb je een oudere computer, dan kun je proberen om een lagere versie te installeren.

Game Maker 6 is gratis te downloaden van de website www.gamemaker.nl. Ga naar de rubriek *Downloads* en kies voor de *Free Version*. Klik op de juiste link en het downloaden start automatisch.

Eens het programma gedownload is, kun je het installeren door te dubbelklikken op het gedownloade bestand gmaker.exe. Volg de instructies op je scherm.

In de laatste stap van het installatieproces, kun je het programma laten starten. Bij het opstarten wordt gevraagd of je de advanced of de beginnersversie wilt starten. Klik op *YES*, zodat de advanced mode wordt ingesteld.

Je kunt het programma later ook starten via de START-knop van Windows en het juiste programma te kiezen in de groep Programma's.

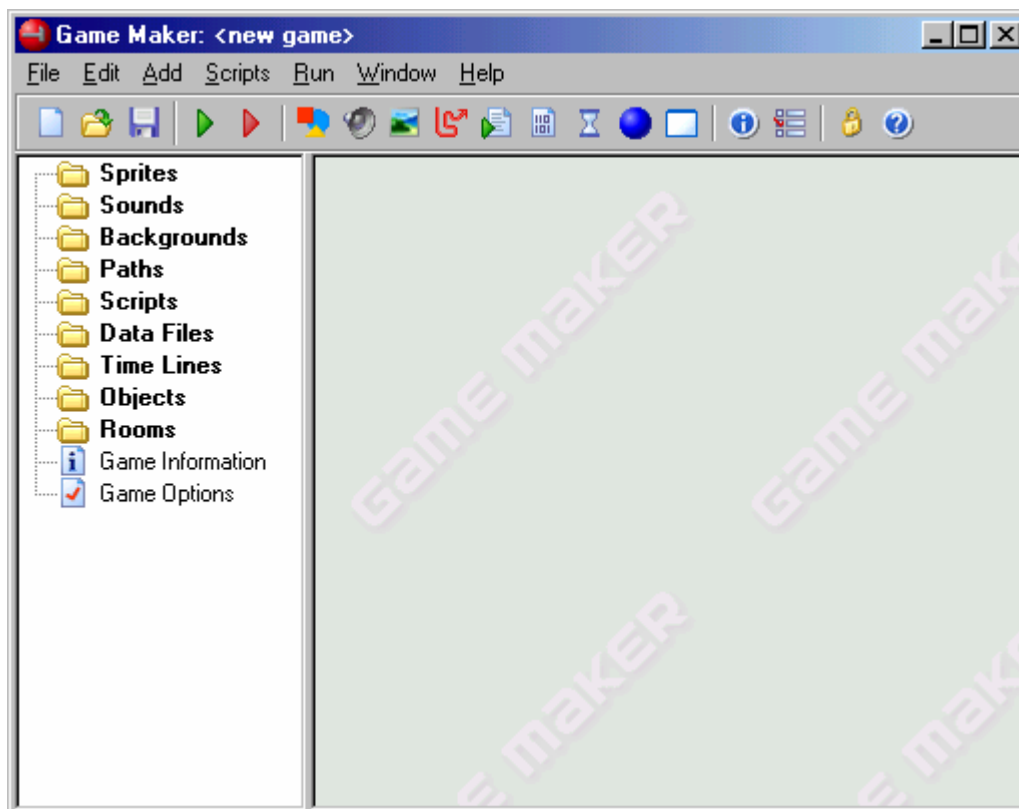
Tijdens de installatie wordt op je harde schijf de map C:\Program Files\Game_Maker6 aangemaakt. Zoek deze map op en bestudeer even wat je aantreft in deze map:

- **examples:** bevat een aantal voorbeelden van spellen die je kunt gebruiken en veranderen, bedoeld als oefeningen.
- **sprites:** bevat een verzameling freeware (animated) sprites die in je spellen gebruikt mogen worden. Een sprite is een afbeelding.
- **backgrounds:** bevat een aantal freeware achtergrondafbeeldingen die je in je spellen kunt gebruiken.
- **sounds:** bevat een aantal freeware geluidseffecten die je in je spellen kunt gebruiken.

Je bent nu klaar om je eerste spel te maken.

De gebruikersinterface

We verklaren de verschillende basisbegrippen van Gamemaker door de gebruikersinterface systematisch te overlopen:



- **Sprites:** dit zijn de afbeeldingen van personages of voorwerpen die opduiken in je spel. Het zijn tekeningetjes: gifs, animated gifs,
- **Sounds:** Geluiden kunnen in spellen zowel worden gebruikt als achtergrond als ook voor effecten, bv. bij het stukschieten van de vijand.
- **Backgrounds:** de afbeeldingen die gebruikt worden als achtergrond op het scherm.
- **Objects:** dit zijn de echte entiteiten in een spel: elk personage, vliegtuig, kanon, ... moet aangemaakt worden als een object. Aan een object kan een sprite worden gekoppeld, zodat het voorgesteld wordt door een mooi figuurtje op het scherm. Een object kan heel wat handelingen doen (bewegen, springen, schieten, botsen...). Er is dus een belangrijk verschil tussen een sprite en een object.
- **Rooms:** de ruimte waarin de objecten leven, bewegen, ... Verschillende levels van je spel, kunnen voorgesteld worden door verschillende rooms.
- **Game Information:** Hier leg je uit hoe je spel in z'n werk gaat. Het is de informatie die verschijnt als je tijdens het spel op F1 drukt.

De overige items (Paths, Scripts, Data Files, ...) komen aan bod als je de basisbegrippen goed onder de knie hebt.

Met de werkbalk kun je de meest gebruikte commando's uitvoeren:



Overloop de verschillende knoppen met de muis zonder er op te klikken: de verklaring van elke knop verschijnt dan op het scherm.

We sommen de belangrijkste knoppen op:



De vertrouwde knoppen: NIEUW – OPENEN - OPSLAAN



Knop om je spel te starten



Knop om een nieuwe sprite toe te voegen.



Knop om een nieuw geluid toe te voegen.



Knop om een nieuwe achtergrond toe te voegen.



Knop om een nieuw object toe te voegen.



Knop om de ruimte te definiëren.

Je eerste spel

Omschrijving & analyse

Vooraleer je aan een programma begint, hoor je een analyse te maken van het probleem en een omschrijving van het programma op te schrijven. Bij een computerspel is dat niet anders. Als je steeds zo te werk gaat, dan bespaar je je later veel tijd.

Het spel dat we gaan maken is erg eenvoudig:

Vang de bal

Een bal botst tegen muren die in een rechthoek zijn geplaatst. Het is de bedoeling dat je de bewegende bal met de linker muiskop aanklikt. Elke keer als dat gelukt is wordt er een geluid geproduceerd, bijvoorbeeld van een geweeschot, en wordt de score opgehoogd met 5. Bij het botsen tegen de muur klinkt een geluidje. Je kunt het spel afsluiten door op ESC te drukken.

Het spel klinkt terecht weinig spectaculair, maar we beginnen eenvoudig. Later kun je snuffjes toevoegen om het spel interessanter te maken. Merk op dat je in de omschrijving ook aangeeft met welke knoppen van muis of klavier het spel gespeeld wordt.

Nu we de omschrijving hebben, maken we een analyse van de verschillende onderdelen van het spel:

Analyse

- *sprites* (2): één voor de muur en één voor de bal.
- *sounds* (2): een geluid voor het aantikken van de bal en een geluid bij het aanklikken met de muis.


- *background* (1): een achtergrond voor de ruimte waarin de bal beweegt.
- *objects* (2): één voor de bal en één voor de muur.
- *rooms* (1): een ruimte waarin de bal beweegt.

We gaan nu stapsgewijs te werk:


- we maken sprites, geluiden en achtergrond aan
- we maken de objecten en de omgeving van het spel klaar
- we leggen de gebeurtenissen en handelingen van het spel vast

De sprites, geluiden en achtergrond aanmaken


Laten we allereerst de sprites maken:

1. Klik op . Er verschijnt een venster. Vul in het veld **Name** "spr_wall" in.
2. Kies vervolgens de **Load Sprite** knop en kies een geschikte afbeelding (die kun je onder andere vinden in de map C:\Program Files\Game_Maker6\Sprites\maze). Klik op OK om het venster te sluiten.
3. Op dezelfde manier als boven beschreven kun je een "ball"-sprite maken.

Nu voegen we de geluiden toe:


4. Klik op . Een ander venster verschijnt nu in beeld. Geef het geluid dat je kiest de naam "snd_beep1" en kies daarna **Load Sound**. Kies een bijpassend geluid uit de map C:\Program Files\Game_Maker6\Sounds. Om te controleren of het inderdaad geschikt is, kun je op de "play"-knop drukken. Als je het geschikte geluid gevonden hebt, kun je het venster sluiten door te klikken op OK.
5. Voeg nu een tweede geluid toe met de naam "snd_zap".

We kiezen nu een achtergrond:



6. Klik op . Maak een achtergrond met de naam "back_blue". Klik nu op Load Background en kies in de map C:\Program Files\Game_Maker6\Backgrounds een mooie achtergrond. Bevestig met OK.

De objecten en de ruimte aanmaken

Nu willen we de 2 objecten gaan maken.

7. Allereerst het object "wall". Klik nu . Er verschijnt weer een venster maar dat zit ietwat ingewikkelder in elkaar. Aan de linkerkant kun je enige algemene informatie over het object vinden.
8. Geef het object de naam "obj_wall" en kies uit het "drop down-menu" de juiste "wall"-sprite. Aangezien een muur onbeweeglijk is, moet je een vinkje zetten in de box **Solid**.
9. Op een zelfde manier maken we een tweede object en noemen dat "obj_ball", kies daarbij de juiste sprite en maak de "ball" niet solid: we gaan dadelijk aan de bal enig gedrag toekennen. Hoe dat precies gebeurt, komt dadelijk aan bod.

Tot slot creëren we de ruimte waar het spel zich afspeelt.

10. Klik nu op . In het venster zie je links verschillende tabblades: kies hier voor SETTINGS en geef de ruimte de naam "room_basic". De afmetingen van de ruimte zijn standaard 640 pixels breed bij 480 pixels hoog. Dat kun je zo laten staan.
11. Kies bovenaan het venster voor een grid van 24 bij 24: vul achter SNAP X en SNAP Y het getal 24 in. De afmetingen van het rooster zijn nu 24 bij 24 pixels.
12. Klik nu op het tabblad BACKGROUNDS en vink DRAW BACKGROUND COLOR uit. Kies in het keuzevak onder FOREGROUND IMAGE voor de juiste sprite. Merk op dat Tile Hor. en Tile Vert. staat aangevinkt: dat betekent dat de achtergrondtegels steeds herhaald wordt totdat de ruimte volledig vol is.
13. Nu gaan we de muur rond de ruimte plaatsen. Klik hiervoor op tabblad OBJECTS. Links in het midden staat een pop-up menu waaruit je objecten kunt kiezen die je aan de room kunt toevoegen op een door jou te kiezen plaats. Zorg dat in de keuzelijst obj_wall staat aangeduid. Klik nu met de linkermuisknop in het rooster op de plaats waar de muur moet komen: voorzie de muur rondom de ruimte. Maximaliseer het venster of gebruik de schuifbalken indien je de ruimte niet volledig op het scherm krijgt. Met de rechtermuisknop kun je een object weer verwijderen.
14. Tot slot kies je nu voor het object "obj_ball" en geef dit object een willekeurige plaats in de ruimte.
15. Klik op  om je instellingen te bevestigen en het venster te sluiten.

De gebeurtenissen en handelingen vastleggen





Tot nu toe heb je alles klaar gemaakt om het spel te maken: de objecten zijn gemaakt, de tekeningetjes en geluiden zijn aangemaakt, de ruimte is gecreëerd. Dit alles volstaat uiteraard niet om een spel te maken. Je moet nu nog definiëren hoe het spel verloopt: hoe mogen de objecten bewegen, welke toetsen of muisknoppen mag de speler gebruiken om het spel te spelen, wanneer moet er een geluid te horen zijn. Dit alles maak je mogelijk door voor elk object events en actions te definiëren. Vooraleer je daaraan begint, ga je eerst per object na welke gebeurtenissen (**events**) en de bijhorende handelingen (**actions**) voorkomen:

object	gebeurtenis (event)	handeling (action)
obj_wall	-	-
obj_ball	van zodra de bal gecreëerd wordt	de bal beweegt in een willekeurige richting
obj_ball	bij het botsen van de bal tegen een muur	de bal moet wegbotsen en er moet een geluidje te horen zijn.
obj_ball	bij het aanklikken van de bal	de score wordt met vijf verhoogd en er moet een geluidje te horen zijn. De bal moet naar een nieuwe plaats springen en opnieuw bewegen in een willekeurige richting.

Nu je precies weet welke events en welke actions er aan elk object gekoppeld moeten worden, gaan we dit ook toevoegen aan de eigenschappen van elk object.

16. Dubbelklik op het object obj_ball. Kies voor **Add Event** en kies voor CREATE. Bij dit event hoort een actie, nl. het bewegen in een willekeurige richting. Deze actie vind je in het tabblad MOVE aan de rechterkant. Sleep deze actie naar de witte zone onder "Actions:". Meteen als je de actie in de lijst hebt gesleept, verschijnt er een nieuw venster waarin je de richtingen van de beweging van het object kunt aangeven. Applies to... laat je

op SELF staan omdat de actie op dit object betrekking heeft. Kies alle 8 de pijlen waarmee je wilt aangeven dat het object alle kanten op kan gaan (random). De speed kun je op 8 zetten. Sluit daarna het venster door te klikken op OK. Nu kan de bal gaan bewegen in een onbepaalde richting zo gauw hij wordt gecreëerd.

17. Een tweede event is de botsing. Kies dus opnieuw voor Add Event en kies voor COLLISION. Je moet meteen een object opgeven waarmee de bal mag botsen. Kies hier voor obj_wall. Bij deze gebeurtenis geef je aan hoe de botsing verloopt. Deze knop vind je in het tabblad MOVE aan de rechterkant. Je hoeft geen instellingen te veranderen: merk op dat de botsing tegen "solid objects" gebeurt en dat we bij het maken van de sprite spr_wall de optie solid hebben aangevinkt. Je kunt dadelijk klikken op OK. Een tweede actie die bij de botsing hoort, is het maken van geluid. Sleep deze knop vanuit het tabblad MAIN1 en vul het scherm dat verschijnt verder aan. Kies hier voor snd_biep1. Je hoeft geen loop te kiezen (false): het geluidje moet zichzelf niet herhalen.  
18. Een derde event is het aanklikken van de bal. Kies weer voor Add Event en kies nu voor MOUSE. Je moet dadelijk aangeven welke knop. Kies voor LEFT BUTTON. Aan dit event koppel je een geluidje op een gelijkaardige manier als hierboven. Kies nu voor snd_zap. Een tweede actie die hierbij hoort is het verhogen van de score. Sleep vanuit het tabblad SCORE de actie "set the score". Geef een waarde van 5 op en klik het vakje "relative" aan. De score moet immers niet op 5 gezet worden: de score moet met vijf verhoogd worden. Een derde actie die hierbij hoort, is het springen naar een willekeurige plaats. Kies in het tabblad MOVE voor "Jump to random position". Je hoeft geen instellingen te veranderen. Voeg nu nog de actie "Start moving in a direction" op dezelfde manier toe als in stap 16.  
19. Alle acties zijn nu ingevoerd. Klik op OK om alle instellingen van obj_ball te bevestigen.

Om een actie achteraf nog te veranderen, dubbelklik je gewoon op de actie.

Je spel testen

Nu is het tijd om ons spel te testen. Testen is heel belangrijk: test je spel zelf en laat het ook door anderen testen. Sla je spel eerst op op je harde schijf onder de naam "bouncing ball".

20. Druk op de **Run**-knop en wacht af wat er gebeurt. Je ziet de score in de titelbalk van het venster. Probeer dan de bal met de muis aan te klikken en kijk dan eens wat er gebeurt. Vraag je af of alles naar wens is: is de beginsnelheid OK, klinken de geluidjes goed, wordt de score juist bijgehouden? Als je geen fouten hebt gemaakt begint de bal rond te bewegen. Het spel is altijd te stoppen door op ESC te drukken. Als iets niet goed is uitgevoerd, heb je de mogelijkheid veranderingen aan te brengen.

21. Gefeliciteerd!! Je hebt nu je eerste spelletje in elkaar gestoken.



Sta nu zelf even stil bij hoe je dit eerste spel hebt aangepakt:

1. je hebt eerst een omschrijving van je spel gemaakt.
2. je hebt daarna een lijst opgesteld van alle sprites, sounds, objecten, ... die in je spel voorkomen. Merk op dat je bij de keuze van de namen steeds een logische naam koos die voorafging door enkele letters die aangeven of het gaat om een sprite, een sound of een object. Je kunt in een naam geen spaties gebruiken, vandaar het gebruik van de underscore.
3. je hebt tot slot alle gebeurtenissen en acties op een rijtje gezet en vertaald naar events en actions. Merk op dat de cruciale objecten (bal en muur) in het spel aanwezig zijn van bij het begin doordat ze in de room geplaatst worden.

Deze werkwijze ga je steeds opnieuw volgen bij het maken van nieuwe spellen.

Je spel verfijnen

De eerste versie van je spel is nu klaar: we gaan het spel nu verder verfijnen.

1. We willen een achtergrondgeluid toevoegen.
2. Om de beweging van de bal niet voorspelbaar te maken, willen we de beweging plots van richting laten veranderen.
3. We willen een uitleg schrijven die verschijnt als je op F1 drukt.
4. We willen van een spel een executable file maken, zodat mensen die niet over Game Maker beschikken het spel ook kunnen spelen.

Achtergrondgeluid toevoegen

Om achtergrondmuziek toe te voegen, moet je beschikken over een MIDI-file. Dit soort muziekbestanden verschillen van de WAV-files die we tot nu gebruikt hebben. Ze zijn te vinden op het internet.

Je gaat als volgt te werk:

1. Maak een nieuw geluid aan met de naam `snd_background` en kies hiervoor een geschikte MIDI-file. Desnoods moet je die eerst even downloaden van het internet, kijk eventueel ook even bij "resources" op de site www.gamemaker.nl.
2. Voeg bij het `obj_ball` bij de creatie van de bal een actie toe: start het geluid `snd_background` en blijf het herhalen (zet dus de instelling "loop" op "true").

Een alarmklok instellen


Om een actie op een willekeurig tijdstip te kunnen laten plaatsvinden, zoals het veranderen van de beweging van de bal, moet je gebruik maken van een alarmklok. Dit betekent dat je bij een object een gebeurtenis ALARM toevoegt waaraan je verschillende acties koppelt. Als het alarm afgaat, worden die acties uitgevoerd op het object.

Je gaat als volgt te werk:

1. Voeg bij het `obj_ball` bij het CREATION event een actie toe: sleep uit het tabblad MAIN2 de actie "set an alarmclock" en zet de "number of steps" op 50. Eens de bal begint te bewegen, begint de alarmklok af te tellen vanaf 50.
2. Om aan te geven wat er gebeurt eens het alarm de nul bereikt, voeg je aan `obj_ball` een event toe: kies voor ALARM. Je moet dadelijk een naam kiezen voor het alarm, kies voor `alarm0`.
3. Aan dit alarm koppel je een actie: sleep uit het tabblad MOVE de actie "start moving in a direction". Je voegt nog een tweede actie toe: je zet de alarmtijd weer op 50. Dat doe je door uit het tabblad MAIN2 de actie "set an alarmclock" te slepen en de "number of steps" weer op 50 te zetten.

Merk op dat je naast de actie "set an alarmclock" ook een event "alarm" moet definiëren. De actie zorgt voor de start van het aftellen. Het event zorgt voor een reeks acties die uitgevoerd worden als het aftellen is afgelopen (als de nul bereikt is).

Een helpmenu toevoegen

Om uitleg toe te voegen aan je spel, klik je gewoon op  Game Information . Je krijgt dan een scherm waarin je tekst kunt ingeven. Met de knoppen bovenaan het scherm kun je opmaakkenmerken instellen. De uitleg verschijnt op het ogenblik dat je op F1 drukt tijdens het spel.

Een executable maken

Om van je spel een executable file (een bestand dat je kunt runnen zonder Game Maker) te maken, ga je als volgt te werk:

1. Zorg dat je spel volledig klaar is en geen fouten bevat.
2. Klik in het menu FILE op CREATE EXECUTABLE.
3. Geef de naam BOUNCING BALL en bevestig met OPSLAAN.

Oefening

Maak nu je spel nog aantrekkelijker:

1. Zoek voor de bal een andere sprite, eventueel eentje die wat groter is of een animated gif die je vindt op het internet.
2. Laat het spel starten met verschillende ballen tegelijkertijd.
3. Laat na elke klik op de bal de snelheid verhogen.
4. Voeg een tweede soort bal toe: als je daarop klikt wordt de score met 10 verlaagd.

Bedenk eventueel zelf nog wat leuke aanpassingen.

Je tweede spel: een scrolling shooter-game

Een scrolling shooter game is een spel waarbij de speler een vliegtuig, ruimteschip of auto bestuurt boven een scrollende achtergrond. Op de achtergrond verschijnen hindernissen die ontweken moeten worden of duiken vijanden op die vernietigd moeten worden. Soms verschijnen ook bonus-objecten die voor extra punten kunnen zorgen.

In dit shooter game met de naam 1945 bestuurt de speler een vliegtuig boven de zee waar vijandelijke vliegtuigen de speler proberen te vernietigen. Dankzij dit spel leer je bewegende achtergronden maken, je vliegtuig besturen, kogels en vijandelijke vliegtuigen sturen. Bij dit spel hoort een scorebord waar onthouden wordt hoeveel levens je nog hebt. Je maakt ook kennis met het gebruik van variabelen, wat uitzonderlijk belangrijk is bij het maken van computerprogramma's.

De omschrijving en de analyse

Vooraleer je aan het programma begint, maak je eerst een ontwerpschema. Dit bevat de omschrijving van het spel, de verschillende sprites en objecten en de verschillende gebeurtenissen en bijhorende handelingen.

1945

In dit spel bestuur je een vliegtuig dat boven een zee vliegt. Je ontmoet vijandige vliegtuigen, die steeds toenemen in aantal. Die vliegtuigen proberen je te vernietigen. Je moet deze vliegtuigen ontwijken of stuk schieten. Het doel van het spel is in leven blijven en zoveel mogelijk vliegtuigen afschieten zodat je een zo hoog mogelijke score hebt. De speler krijgt drie levens: als de drie levens op zijn, explodeert het vliegtuig van de speler en verschijnt de scoretabel. Eén leven van de speler is vernietigd als het vier vliegtuigen heeft geïncasseerd.

Op F1 drukken levert een help-menu en op ESC drukken eindigt het spel. De spatiebalk dient om te schieten.

analyse

- *sprites (10)*: het vliegtuig van de speler, het vliegtuig van de vijand dat de speler vernietigt als het vliegtuig van de vijand raakt, een kogel, drie types eilanden in de zee, twee soorten explosies, de achtergrond van het scorebord, een voorstelling van een leven op het scorebord.
- *sounds (2)*: een geluid bij het stukschieten van de vijand, een geluid bij een explosie van het vliegtuig van de speler.
- *background (1)*: de voorstelling van de zee op de achtergrond
- *objecten (10)*: het vliegtuig van de speler, het vliegtuig van de vijand, de kogel, de drie eilanden, de twee explosies, een object dat vijandige vliegtuigen creëert, een object dat de score en het aantal levens bepaalt.
- *rooms (1)*: de bewegende achtergrond, de illusie van de zee die onder het vliegtuig doorschuift

De gebeurtenissen en acties

Typisch aan dit spel is de bewegende achtergrond. Dit wordt verwezenlijkt door de achtergrond van de room (de zee) een vaste snelheid te geven. Voorwerpen op de achtergrond (de eilanden) zullen dezelfde snelheid krijgen om de illusie te wekken dat ze in die achtergrond liggen.

object	gebeurtenis (event)	handeling (action)
obj_island1 obj_island2 obj_island3	van zodra het eiland gecreëerd wordt	het eiland krijgt dezelfde verticale snelheid als de achtergrond
obj_island1 obj_island2 obj_island3	van zodra het eiland de rand van het spel heeft bereikt	het eiland krijgt een nieuwe willekeurige positie bovenaan het scherm, zodat het als nieuw eiland kan fungeren.

Het vliegtuig van de speler moet kogels kunnen afvuren. Om de speler niet de mogelijkheid te geven voortdurend te schieten, mag de speler slechts twee kogels afvuren per seconde, dat betekent één om de 15 stappen. Om dit mogelijk te maken wordt een variabele ingevoerd met de naam can_shoot. Indien deze variabele de waarde één heeft, kan het vliegtuig schieten, indien deze de waarde nul heeft, kan het vliegtuig niet schieten. Bij de creatie van het vliegtuig van de speler wordt deze waarde op één gezet. Bij het afschieten van een kogel wordt deze waarde op nul gezet en start een alarmklok. Wanneer deze klok op nul komt wordt de waarde can_shoot weer op één gezet.

object	gebeurtenis (event)	handeling (action)
obj_myplane	van zodra het vliegtuig gecreëerd wordt	de variabele can_shoot wordt op één gezet
obj_myplane	duwen van de LEFT-toets	het vliegtuig beweegt 4 stappen naar links, tenzij het tegen de rand zit
obj_myplane	duwen van de RIGHT-toets	het vliegtuig beweegt 4 stappen naar rechts, tenzij het tegen de rand zit
obj_myplane	duwen van de UP-toets	het vliegtuig beweegt 2 stappen naar boven, tenzij het tegen de rand zit
obj_myplane	duwen van de DOWN-toets	het vliegtuig beweegt 2 stappen naar onder, tenzij het tegen de rand zit
obj_myplane	duwen van de spatiebalk	indien de variabele can_shoot op één staat, wordt een kogel afgevuurd, krijgt de variabele de

		waarde nul en start de alarmklok met af te tellen vanaf 15.
obj_myplane	afgaan van het alarm	de variabele can_shoot krijgt waarde één
obj_enemy1	van zodra het vliegtuig gecreëerd wordt	het vliegtuig krijgt een verticale snelheid mee die groter is dan de achtergrond
obj_enemy1	van zodra het vliegtuig de rand van het spel heeft bereikt	het vliegtuig krijgt een nieuwe willekeurige positie bovenaan het scherm, zodat het als nieuwe vijand kan fungeren.
obj_enemy1	indien het vliegtuig botst met het vliegtuig van de speler	je hoort het geluid van een explosie, je ziet een afbeelding van een explosie, het vliegtuig van de vijand springt naar een nieuwe positie bovenaan het scherm, het toestel van de speler loopt schade op
obj_enemy1	indien het vliegtuig botst met een kogel	je hoort het geluid van een explosie, je ziet een afbeelding van een explosie, de kogel wordt vernietigd, het vliegtuig van de vijand springt naar een nieuwe positie bovenaan het scherm, de score van de speler verhoogt met vijf.

Er zijn twee soorten explosies. Een explosie van het vliegtuig van de vijand: die vindt plaats als het vliegtuig geraakt wordt door een kogel of als de vijand tegen het vliegtuig van de speler vliegt. Een tweede explosie is die van het vliegtuig van de speler: die vindt plaats als het een leven van de speler vernietigd heeft: na vier botsingen met de vijand explodeert het vliegtuig van de speler, de afbeelding van het vliegtuig van de speler wordt dan vervangen door een explosie. De speler verliest dan een leven.

object	gebeurtenis (event)	handeling (action)
obj_explosie1	van zodra de explosie gedaan is	de afbeelding van de explosie moet verdwijnen
obj_explosie2	van zodra de explosie gedaan is	de afbeelding van de explosie moet verdwijnen, er moet een nieuw vliegtuig voor de speler gemaakt worden, het aantal levens moet met één verminderen.

Uiteindelijk is er nog de kogel die als object enkele events en acties moet toegewezen krijgen.

object	gebeurtenis (event)	handeling (action)
obj_bullet	van zodra de kogel gecreëerd is	de kogel krijgt een verticale snelheid mee, die tegengesteld is aan de beweging van de achtergrond
obj_bullet	van zodra de kogel de rand van het scherm bereikt heeft	de kogel vernietigt zichzelf en verdwijnt van het scherm

Tot slot zijn er nog twee objecten: het object dat de schade aan het vliegtuig, het aantal levens en de score bijhoudt en het object dat het aantal vliegtuigen van de vijand doet toenemen. Op beide objecten wordt later ingegaan omdat het hier om nieuwe begrippen gaat.

De sprites, de sounds, objects en background

Om de sprites, de sounds en de background van dit spel aan te maken, dien je eerst de nodige bestanden op je harde schijf te zetten. Surf daartoe naar www.gamemaker.nl en ga naar tutorials. Download de tutorial Creating a Scrolling Shooter. Dit is een zip-bestand dat je moet unzippen vooraleer je het kunt gebruiken. Nadat het bestand geunzip is, kun je de sprites, sounds enz. gebruiken voor dit spel.

Maak nu alle sprites, sounds en background aan. Gebruik de volgende bestandsnamen:

spr_island1	snd_explosion1	<i>back_water</i>
spr_island2	snd_explosion2	
spr_island3		
spr_myplane		
spr_enemy1		
spr_bullet		
spr_explosion1		
spr_explosion2		
spr_bottom		
spr-life		

Neem voor explosion1 een kleine explosie en voor explosie 2 een grote explosie.

Bij een aantal sprites definiëren we de oorsprong van de afbeelding. Dubbelklik daartoe op de juiste sprite en geef bij ORIGIN de juiste coördinaten op:

bij spr_myplane:	X:32	Y:32
bij spr_bullet:	X:16	Y:16
bij spr_explosion1:	X:16	Y:16
bij spr_explosion2:	X:32	Y:32

Explosies van vliegtuigen van de vijand kunnen vaak tegelijkertijd voorkomen. Stel daarom bij snd_explosion1 de buffer in op 4: daardoor kunnen vier van deze geluiden tegelijkertijd afgespeeld worden.

Maak voor elke sprite ook een object, voorlopig nog zonder de events en actions toe te kennen. We definiëren bij de objecten nu ook een diepte (depth): de objecten worden getekend in volgorde van diepte. Afbeeldingen van objecten met de grootste diepte worden het eerst getekend en liggen het laagst op de tekening. Gebruik de volgende bestandsnamen:

```
obj_island1 (diepte 10 000)
obj_island2 (diepte 10 000)
obj_island3 (diepte 10 000)
obj_myplane (diepte -100)
obj_enemy1 (diepte 0)
obj_bullet (diepte 0)
obj_explosion1 (diepte 0)
obj_explosion2 (diepte 0)
```

Tot slot maken we de bewegende achtergrond. We gaan als volgt te werk:

1. Maak een nieuwe room met de naam `room_sea`. We houden de standaard settings: de `room_width` is 640, de `room_height` is 480.
2. Verander de achtergrond in `back_water`.
3. Zorg dat `Tile Hor.` en `Tile Vert.` aangevinkt zijn.
4. Zet de `Vert. Speed` op 2.
5. Voeg nu op een willekeurige plaats de drie objecten van de eilanden toe.
6. Voeg ergens centraal het vliegtuig van de speler toe.
7. Voeg ergens bovenaan het vliegtuig van de vijand toe.

Het grote kader van het spel is nu klaar. Nu kunnen we starten met stapsgewijs de gebeurtenissen en opdrachten toe te voegen. Alvorens we daar echt mee verder kunnen, moeten we even iets over variabelen zeggen.

Het gebruik van variabelen

Een variabele is een eigenschap van een instantie van een object. Merk op dat het programma Game Maker een onderscheid maakt tussen een object en een instantie van een object. Met instantie bedoelt men een weergave van het object. Van één object kunnen immers meerdere weergaven aanwezig zijn in het spel, met allemaal dezelfde eigenschappen die vastgelegd zijn in het object. (vb. de vijandelijke vliegtuigen in dit spel).

Bij het definiëren van een object kun je verschillende eigenschappen vastleggen:

<code>x</code>	de x-coördinaat van een instantie
<code>y</code>	de y-coördinaat van een instantie
<code>hspeed</code>	de horizontale snelheid (in pixels per stap)
<code>vspeed</code>	de verticale snelheid (in pixels per stap)
<code>solid</code>	of een object solid (1) is of niet (0)

Al deze eigenschappen noemt met **locale variabelen**: ze zijn enkel van toepassing op het object waarvoor ze gedefinieerd worden. Om bijvoorbeeld de x-coördinaat van het object `obj_myplane` aan te geven gebruik je de twee benamingen, gescheiden door een punt: vb. `obj_myplane.x`

In een computerprogramma zijn er ook **globale variabelen** mogelijk, die onafhankelijk zijn van een object.

<code>score</code>	de huidige score
<code>lives</code>	het aantal levens op dit ogenblik
<code>mouse_x</code>	de x-positie van de muis
<code>mouse_y</code>	de y-positie van de muis
<code>room_width</code>	de breedte van de room in pixels
<code>room_height</code>	de hoogte van de room in pixels

Naast deze ingebouwde variabelen, kun je ook je eigen variabelen invoeren. Zoals we verder zullen zien, voeren we een variabele `can_shoot` in die twee waarden kan hebben: indien die nul (false) is, kan de speler niet schieten, indien die één (true) is wel. Bij het kiezen van namen voor variabelen moet je er rekening mee houden dat een naam hoofdlettergevoelig is. `Can_Shoot` is niet hetzelfde als `can_shoot`. Merk op dat de variabelen die je zelf invoert altijd lokaal zijn en enkel horen bij het object waarvoor ze zijn gecreëerd. Zorg ervoor dat ze een verschillende naam hebben van reeds aanwezige variabelen. Dat kun je realiseren door de namen vooraf te laten gaan door `var_`.

Wil je je eigen globale variabelen maken dan moet je laten voorafgaan door `global` gevolgd door een punt: vb. `global.bonus`

De handelingen die je met variabelen kunt uitvoeren zijn te vinden in het tabblad `CODE`.



Set the value of a variable

Met deze actie kun je de waarde van een variabele aanpassen. Je geeft dan naam van de variabele op en de nieuwe waarde. Deze nieuwe waarde kan ook een expressie (een formule) zijn. Wanneer je `Relative` aanduidt, betekent dat de nieuwe waarde bij de oorspronkelijke waarde wordt bijgeteld. Merk op dat dat betekent dat de variabele dan wel al eerder een waarde moet toegekend zijn, anders zal er een fout optreden.



If a variable has a value

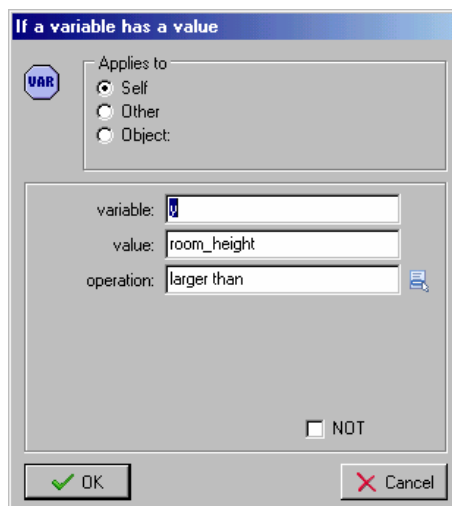
De acties die hierop volgen worden bepaald door de uitkomst van deze IF-test. Indien deze de waarde `true` oplevert worden de acties uitgevoerd. Indien deze de waarde `false` oplevert, worden de acties niet uitgevoerd.

Het toekennen van gebeurtenissen en opdrachten

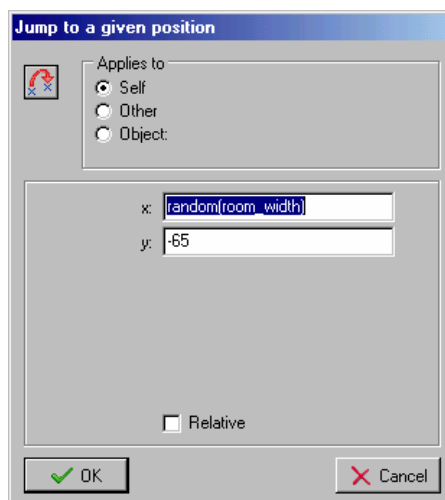
We gaan stap voor stap te werk en volgen daarbij het schema dat we hoger hebben uitgeschreven. We beginnen dus met de drie eilanden.

De eilanden

1. Dubbelklik op `obj_island1` en kies voor `Add Event`. Kies voor `CREATE` en sleep vanuit het tabblad `MOVE` de actie "set vertical speed". We kiezen dezelfde snelheid als de achtergrond: 2. Om aan te geven dat het eiland bij het naderen van de rand van de "room" naar een nieuwe locatie moet springen, kiezen we een nieuw event, `STEP`. We voegen een actie toe die de waarde van de `y`-coördinaat bij elke stap controleert. Indien deze groter wordt dan `room_height`, moet het eiland naar een nieuwe willekeurige positie springen, buiten beeld. Sleep hiertoe vanaf het tabblad `CODE` de actie "If a variable has a value" naar het actieveld. Onderstaand dialoogkader verschijnt. We doen een test op de `y`-coördinaat, indien die groter wordt dan de `room_height`, moet een actie uitgevoerd worden. Stel het venster in en klik op `OK`.



Kies vanuit het tabblad MOVE de actie “Jump to a given position” en stel het venster in zoals hieronder:

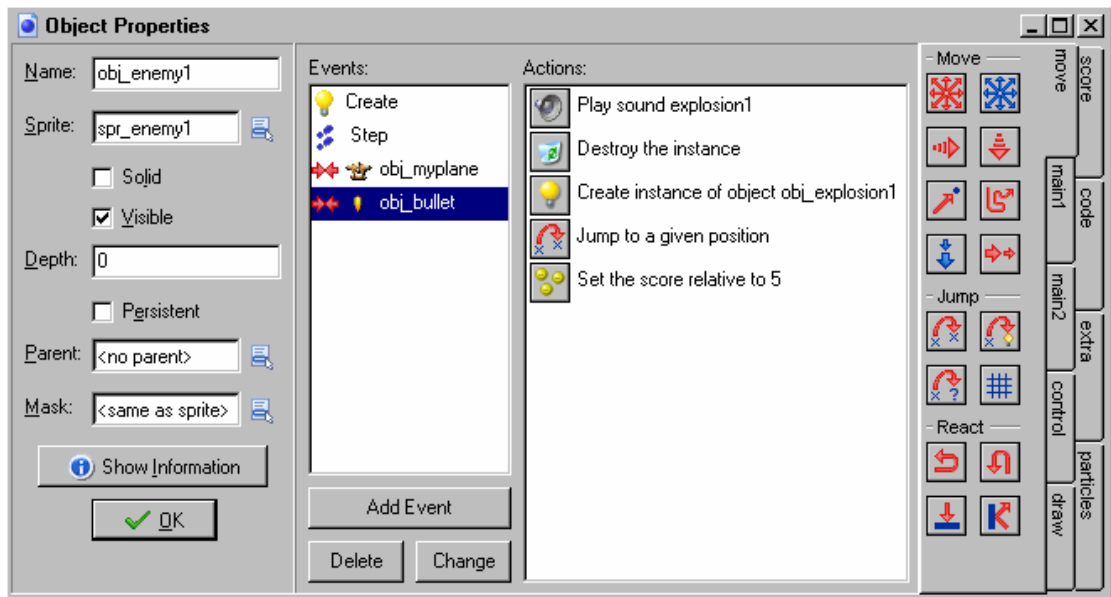


Dit betekent dat de nieuwe y-coördinaat -65 wordt (buiten beeld dus) en de nieuwe x-coördinaat is een willekeurig getal tussen nul en de `room_width`. Doe de juiste instellingen en klik op OK.


- Herhaal deze bewerkingen voor de andere twee eilanden.


Het vliegtuig van de vijand

- Voor het `obj_enemy1` doen we net hetzelfde als bij de eilanden, alleen stellen we de verticale snelheid in op 4, dus groter dan de snelheid van de achtergrond en de eilanden. Ook de positie waar het vliegtuig naartoe springt na het bereiken van de rand mag anders zijn: kies voor de y-coördinaat -16 . De x-coördinaat blijft willekeurig.
- We voegen nu nog twee events toe: een botsing met een kogel en een botsing met het vliegtuig van de speler. Maak eerst de event “botsing met een kogel”. De bijhorende acties zijn al beschreven in de tabel die hoger werd opgesteld. We vatten ze samen door onderstaande schermafdruk:

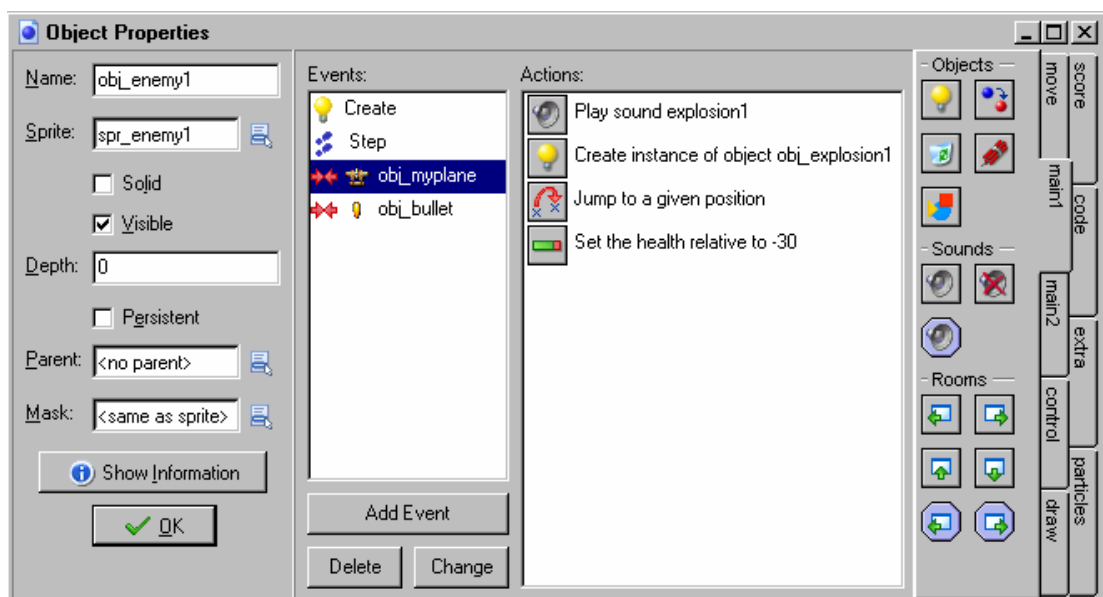



De nieuwe acties  &  vind je in het tabblad MAIN1:

 Hiermee vernietig je de kogel. Daartoe kies je bij “Applies to:” voor other. Aangezien deze actie bij het vliegtuig van de vijand hoort, is het “andere object” de kogel.

 Hiermee creëer je een explosie. Kies bij object voor “obj_explosion1” en bij de positie kies je (0,0). Vink ook de optie “relative” aan: daardoor komt de explosie precies op de plaats van het vliegtuig van de vijand in beeld.

We stellen nu de gebeurtenis “botsing met obj_myplane in”. De verschillende acties worden hieronder samengevat:



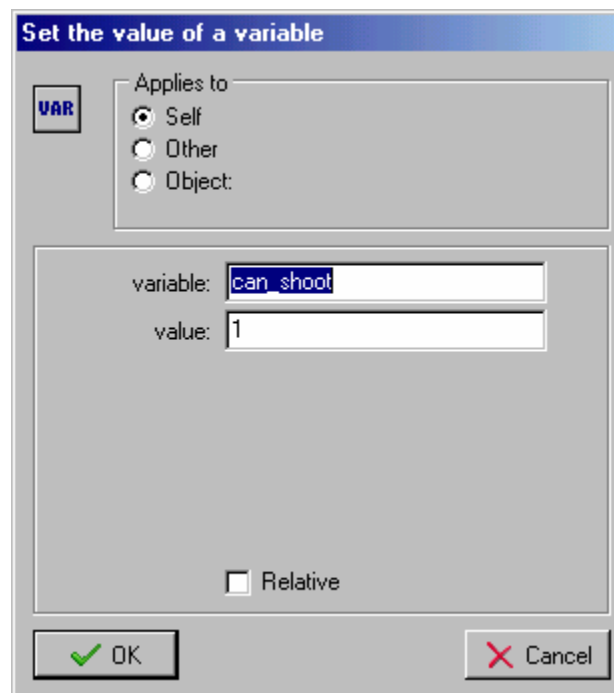
De nieuwe actie  vind je in het tabblad SCORE: met deze actie kun je bijhouden hoe zwaar het leven van de speler is toegenomen. Als de waarde van de variabele health op nul komt, verdwijnt er één leven van de speler. Kies als value -30 en klik de optie “relative”

aan. Bij elke botsing verdwijnen dus 30 punten op een schaal van 100. Na de vierde botsing zal dus een leven moeten verdwijnen.

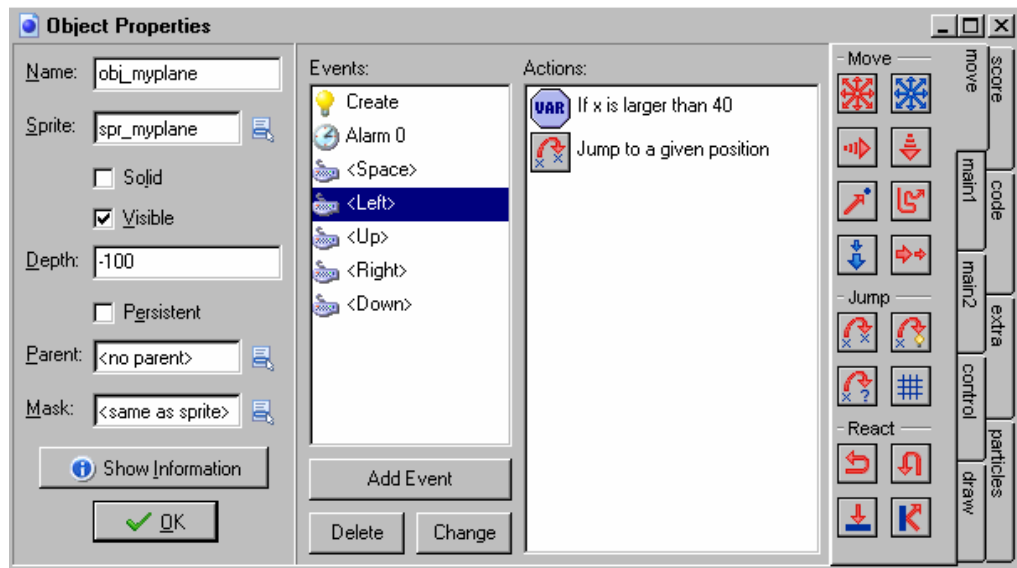
Het vliegtuig van de speler

Nu definiëren we de events en actions voor obj_myplane. Volg hier opnieuw de opsomming die we hoger in de tabel hebben uitgewerkt.

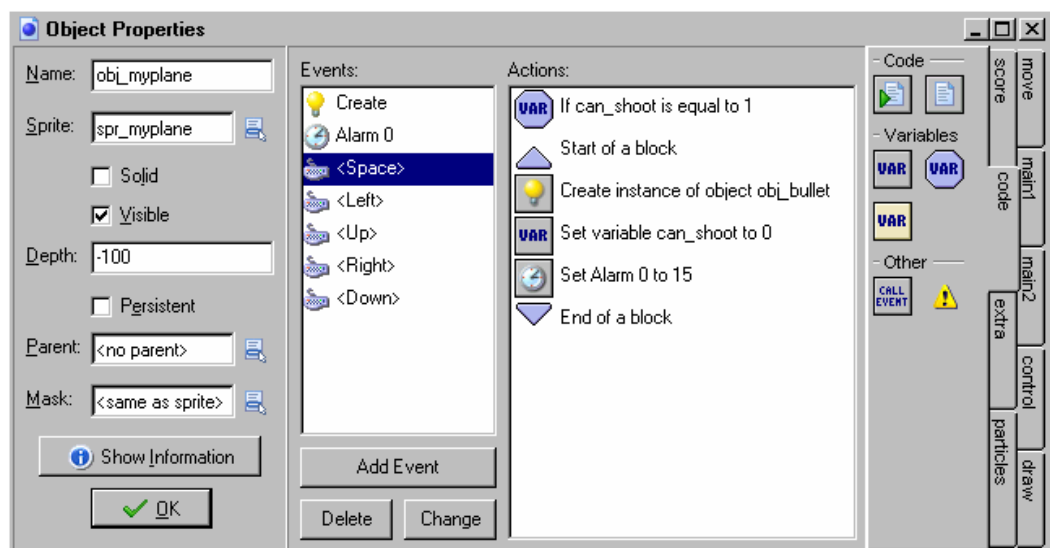
1. Voeg de event CREATE toe en sleep de actie “set the value of a variable” vanuit het tabblad CODE naar het action-veld. Geef aan dat de variabele can_shoot op 1 moet worden gezet.



2. We bepalen nu de invloed van de pijltoetsen op het vliegtuig. Voeg het event KEYBOARD toe en kies voor LEFT. Deze toets moet zorgen dat het vliegtuig vier plaatsen naar links beweegt. We willen dat dit niet mogelijk is als het vliegtuig al helemaal links staat. Daarom voegen we een IF-test toe. Denk hierbij aan de opbouw van het scherm. De oorsprong (0,0) van het assenstelsel ligt links-boven. We testen de x-coördinaat: enkel indien die groter is dan 40 laten we het vliegtuig naar links bewegen. Sleep dus vanuit het tabblad CODE de actie “If a variable has a value” en kies als variabele x. Pas de andere instellingen aan en klik op OK. Onder deze actie moet nu opgesomd worden wat er dient te gebeuren als aan de voorwaarde is voldaan: sleep de actie “jump to a given position” vanuit het tabblad MOVE. Geef aan dat de x-coördinaat -4 wordt en de y-coördinaat 0, relatief ten opzichte van de huidige positie. Dit zal betekenen dat de LEFT toets het vliegtuig inderdaad naar links doet gaan.



3. Herhaal nu de bovenstaande werkwijze voor de UP, RIGHT en DOWN-toets:
 - UP: If y larger than 40, jump to (0,-2)
 - RIGHT: If x smaller than room_width, jump to (4,0)
 - DOWN: If y smaller than room_height-120, jump to (0,2)
4. Merk op dat elke positie relatief wordt bekenen, ten opzicht van de vorige. Bij de DOWN-bewegiging blijven we een afstand van 120 pixels van de rand omdat we op die plaats het scorebord willen plaatsen.
5. Nu geven we aan wat er dient te gebeuren als er op de spatiebalk wordt geduwd. Herinner je dat de speler niet constant mag vuren. Daarom wordt een variabele `can_shoot` ingevoerd. Wanneer die op één staat is vuren toegestaan. Alvorens te kunnen vuren, moet dat getest worden via een IF-test. Voeg het event KEYBOARD toe en kies voor SPACE. Vul dan de bijhorende acties aan. Merk op dat indien een IF-test meerdere commando's tot gevolg heeft, alle commando's tussen haken worden geplaatst. Je vindt deze in het tabblad CONTROL.

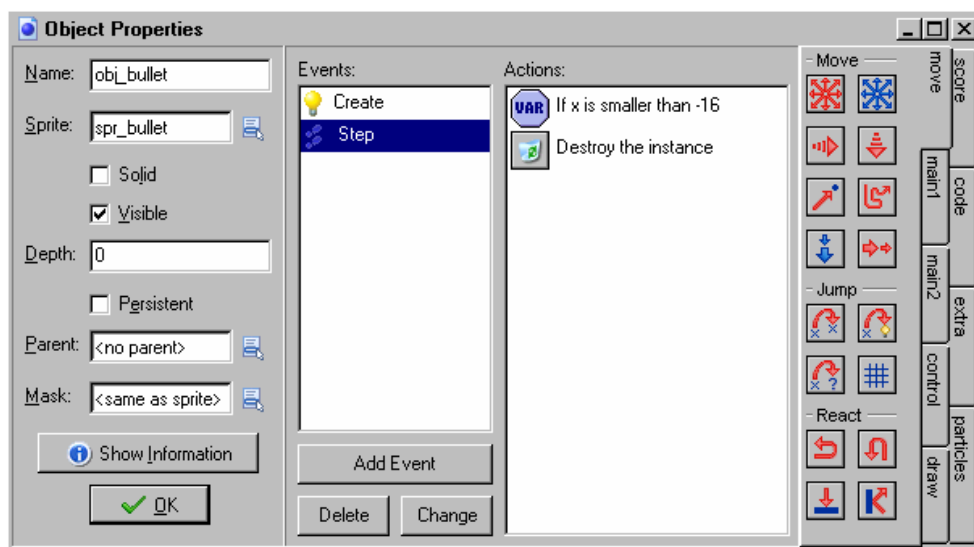


6. Je hebt net een alarmklok toegevoegd die aftelt van zodra een kogel wordt toegevoegd. Dat betekent dat je ook een event ALARM moet bepalen die beschrijft wat er dient te gebeuren als het alarm af gaat. Voeg dus een nieuw EVENT ALARM toe. De bijhorende actie zet de variabele `can_shoot` weer op één.

De kogel

We stellen nu de events en actions van obj_bullet in, zodat we het spel dadelijk al een eerste keer kunnen testen.

7. Kies het event CREATE. Geef bij de creatie de kogel een verticale snelheid van -8 . Dit is dus tegengesteld aan de snelheid van de achtergrond.
8. Om ervoor te zorgen dat de kogel zichzelf vernietigt als die de rand van het spel bereikt, testen we bij elke stap de positie. Indien de y-coördinaat kleiner wordt dan -16 (waarom -16 ?), vernietigt de kogel zichzelf. Voeg een event STEP toe. Koppel hieraan een actie "If a variable has a value", kies als variabele y en kies als de actie die hierop volgt "destroy the instance".



De eerste test

Om te testen hoe ver het nu staat met je spel, kun je het even runnen en uitproberen of alles tot nu toe goed loopt. Test de beweging van het vliegtuig van de speler, ga eens tot tegen de rand, probeer een vijandig vliegtuig af te schieten. Laat ook een vijandig vliegtuig tegen jouw vliegtuig botsen. Laat dat op z'n minst vier keer gebeuren. Verlaat nu het spel via ESC.

Indien het moederschip niet correct beweegt of bij het afschieten van een kogel loopt iets fout, kun je best eerst de vorige stappen in je programma opnieuw doornemen en de fout zoeken.

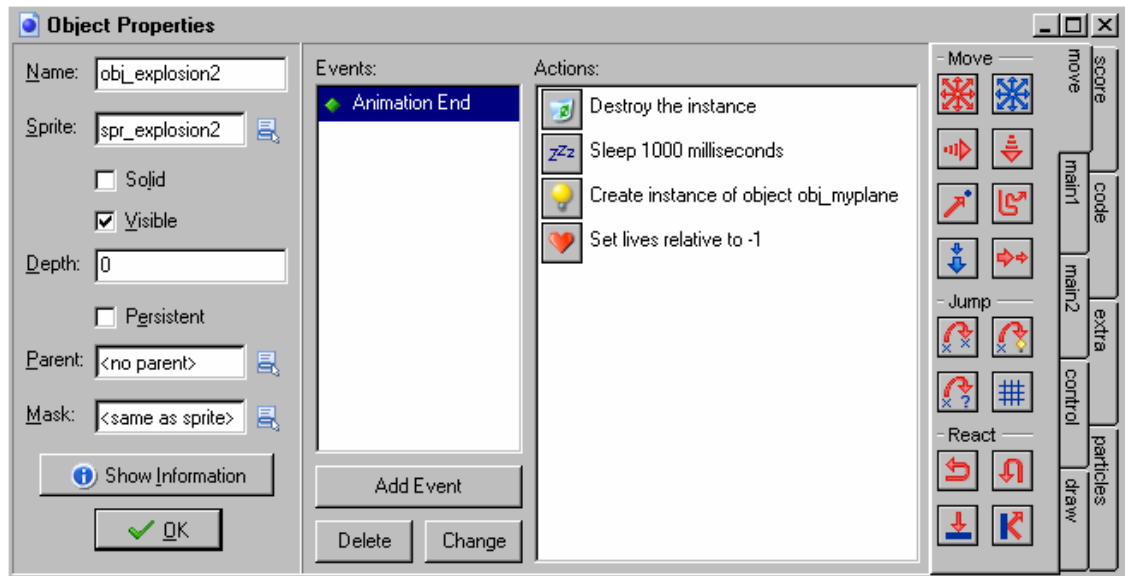
Indien alles naar wens verloopt, stel je vast dat toch niet alles in orde is:


- het moederschip vernietigt zichzelf niet na verschillende botsingen.
- er is nog geen goed scorebord met een overzicht van de levens.
- het aantal vijandige vliegtuigen blijft beperkt tot één.
- de explosies blijven op het scherm.


De explosies

We lossen eerst dit laatste probleem op. Dit is een gevolg van het feit dat we nog geen acties hebben gekoppeld aan obj_explosion1.

1. Obj_exlosion1 is de kleine explosie die plaatsvindt als een vliegtuig van de vijand wordt geraakt. De enige actie die deze explosie moet ondernemen is zichzelf vernietigen als de explosie voorbij is. Hiervoor bestaat een speciaal event dat je vindt onder OTHER, kies voor ANIMATION END. De actie die je hieraan koppelt is "Destroy the instance".
2. We stellen meteen ook Obj_exlosion2 in. We gaan gelijkaardig te werk, maar voegen nog enkele acties toe. We vervangen de explosie door een nieuw vliegtuig voor de speler en verlagen het aantal levens met één.



De actie  heeft als doel het spel even te onderbreken gedurende 1 seconde, zodat het geluid van de explosie zeker voorbij is, alvorens een nieuw vliegtuig aan te maken. Je vindt deze actie in het tabblad MAIN2.

De actie  zorgt ervoor dat je het aantal levens kunt instellen en veranderen. Kies hier voor een waarde -1, relatief genomen.

Test je spel eventueel een tweede keer en bekijk het effect van deze laatste aanpassingen. Nog steeds heeft het aantal botsingen van de vijand met het vliegtuig van de speler geen invloed op het leven de speler. Daarvoor is een bijkomend object nodig.

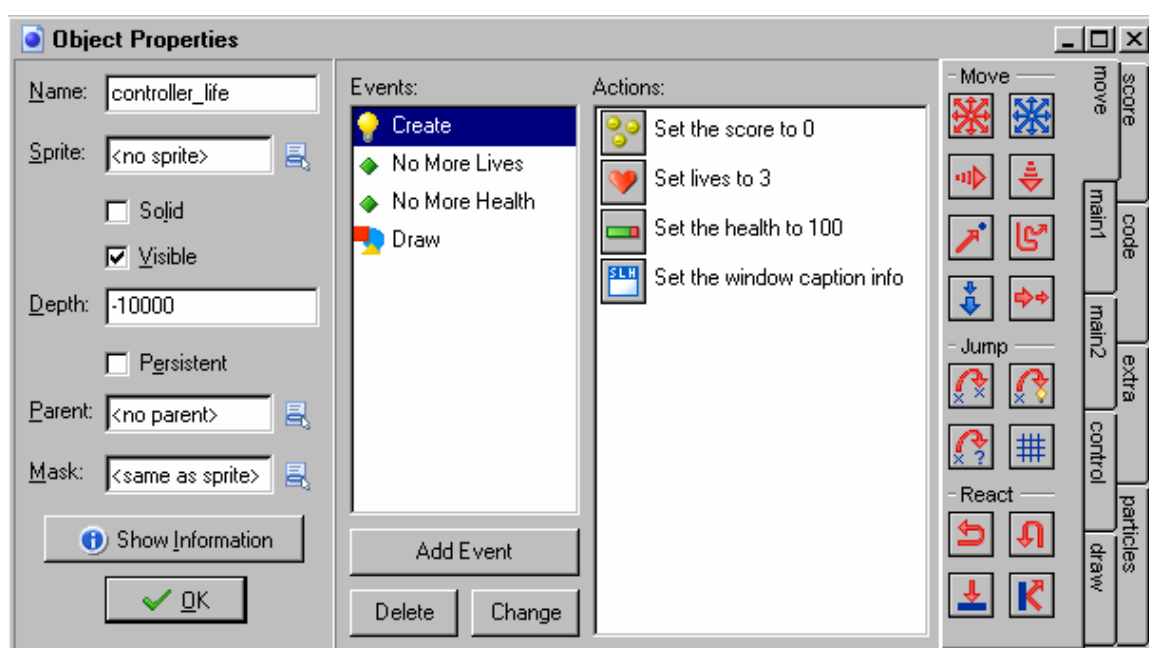
Het scorebord

Het tonen van een scorebord, het bijhouden van het aantal levens en de toestand van het vliegtuig van de speler (health) wordt allemaal gecontroleerd door één nieuw object. Creëer hiervoor het object controller_life. Aan dit object wordt geen sprite gekoppeld. Om er zeker van te zijn dat dit object als eerste wordt gecreëerd, geef je het een depth-waarde van -10000.

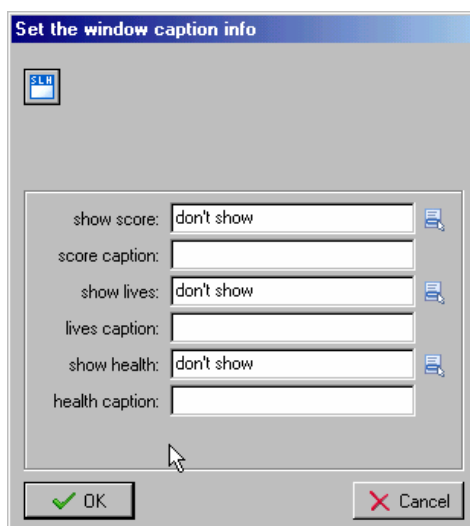
object	gebeurtenis (event)	handeling (action)
controller_life	van zodra de controller gecreëerd is	de score wordt op nul gezet, het aantal levens op drie, de health op 100.
controller_life	teken op het scherm de bijhorende	teken de sprite van het scorebord,

	waarden	stel een lettertype in voor de score, geef de score weer, geef de health-waarde weer, geef het aantal levens weer
controller_life	indien de health op nul staat	het vliegtuig van de speler explodeert en wordt vervangen door een explosie, je hoort het bijhorende geluid, de health wordt weer op 100 gezet
controller_life	indien het aantal levens op nul staat	toon de high-score tabel en start het spel opnieuw

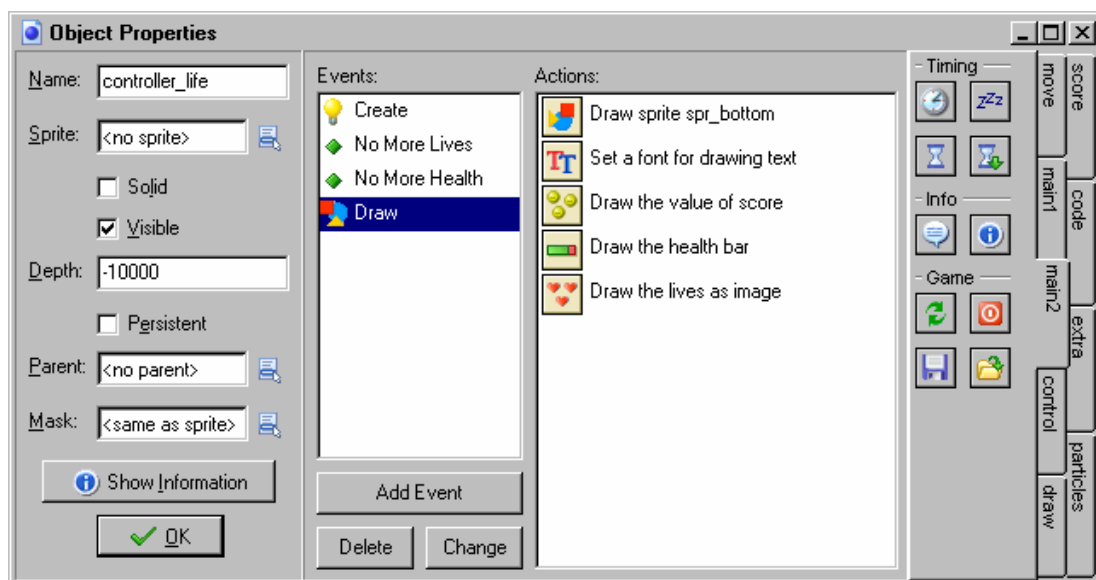
We maken nu de events en acties aan uit bovenstaande tabel:





1. Voeg het event CREATE toe en koppel hieraan de bijhorende acties uit het tabblad SCORE. Merk op dat een vierde actie wordt toegevoegd "set the window caption info". Deze actie is nodig om te vermijden dat bij de score, lifes of healths overbodige tekst wordt weergegeven.



2. We voegen nu een nieuw event DRAW toe. Dit event maakt het mogelijk om alle waarden van levens, score en health te tonen op het scherm.



3. De nieuwe acties  en  vind je in het tabblad DRAW. De overige acties vind je in het tabblad SCORE. Merk het verschil op met de reeds hoger gebruikte acties: het gaat hier om draw-acties met een icoon met witte achtergrond.



Met deze actie tekenen we het scorebord, dat bewaard is in spr_bottom. We nemen als x-waarde 0 en als y-waarde 404. Dat betekent dat de oorsprong (origin) van de sprite op de positie (0,404) wordt geplaatst. De hoogte van deze sprite is immers 76, zodat de ruimte tussen y=404 en y=480 volledig opgevuld wordt met het scorebord. Het subimage staat standaard op -1. Deze variabele heeft nu geen betekenis omdat de sprite maar uit één image bestaat.



Kies via actie een mooi lettertype voor je score, kies ook een licht kleur voor je lettertype.



Hiermee teken je de score. Kies als coördinaat (180, 440) en vul niets in bij caption. Dit betekent dat er geen tekst wordt weergegeven bij de score.

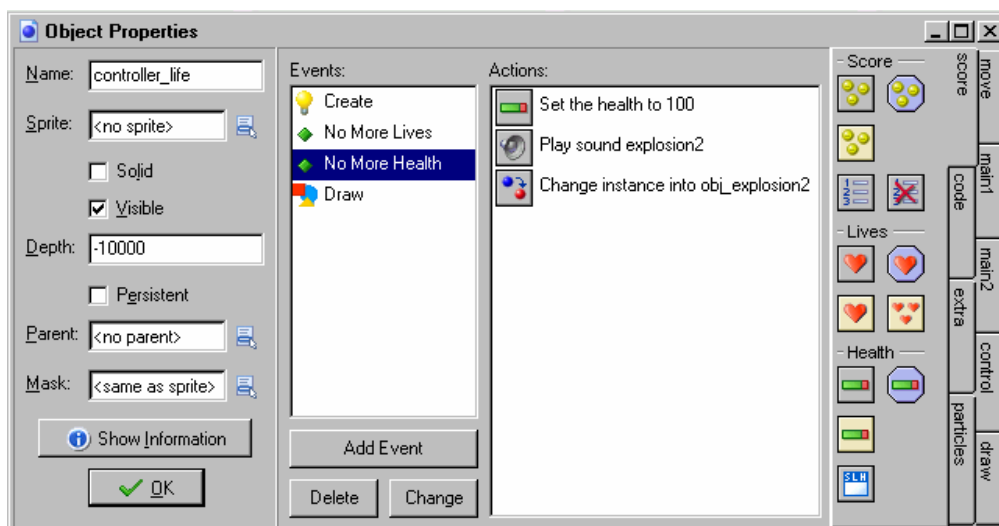


Hiermee stel je de health-waarde voor door een balk die evolueert van groen naar rood. Kies als coördinaat voor de linkerbovenhoek (12,449) en voor de rechteronderhoek (138,459). Dit betekent dat de balk 10 pixels hoog is en 126 pixels breed. Kies nu nog een back-color en stel de bar color in op van green to red.

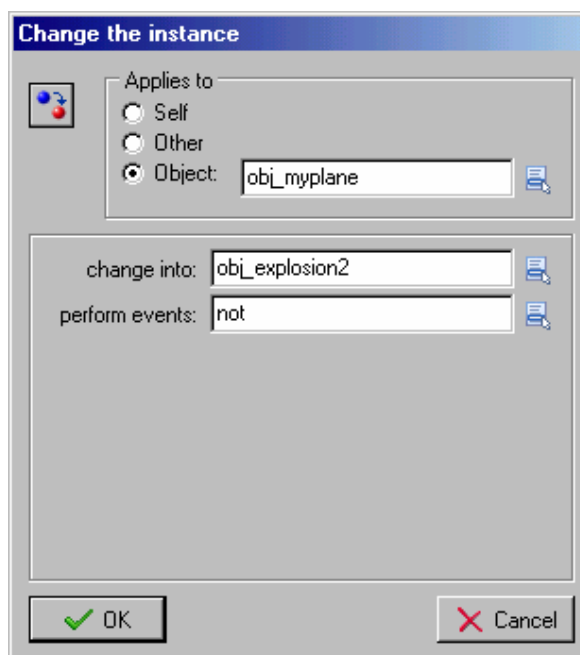


Hiermee maak je je levens visueel. Kies als coördinaat (16,410) en kies als sprite spr_life. Deze sprite zal nu de drie levens voorstellen.

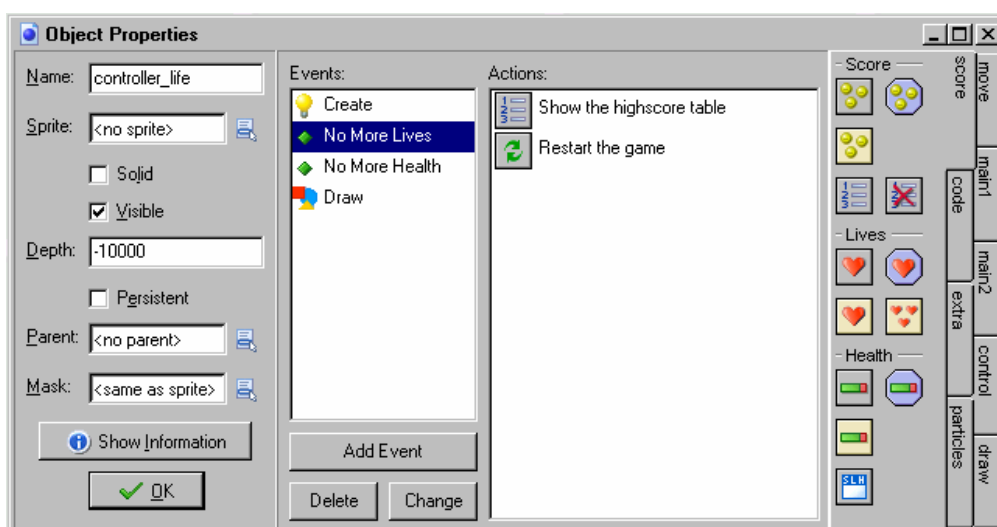
4. Voor het derde event van controller_life maken we kennis met een nieuw event: kies bij OTHER voor NO MORE HEALTH. Herinner je dat health aangeeft hoe erg het vliegtuig van de speler is toetakeld en dat bij elke botsing met een vijandig vliegtuig de health-waarde met 30 wordt verminderd. Stel de nodige acties in.



Opdat na vier vijandige botsingen het moederschap zou vernietigd worden, vervangen we het door een explosie, nl. obj_explosion2. Sleep het commando CHANGE THE INSTANCE naar het actie-scherm. Vul het kadertje aan zoals hieronder aangegeven:



- Als laatste event voegen we NO MORE LIVES toe. We maken hier gebruik van een nieuwe actie uit het tabblad SCORE, nl. “show the high-score table” en een actie uit het tabblad MAIN2.



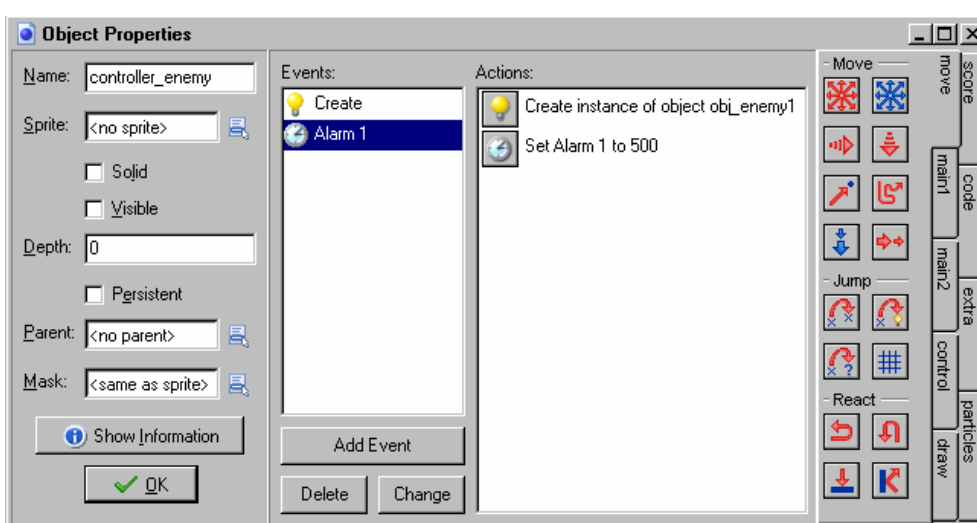
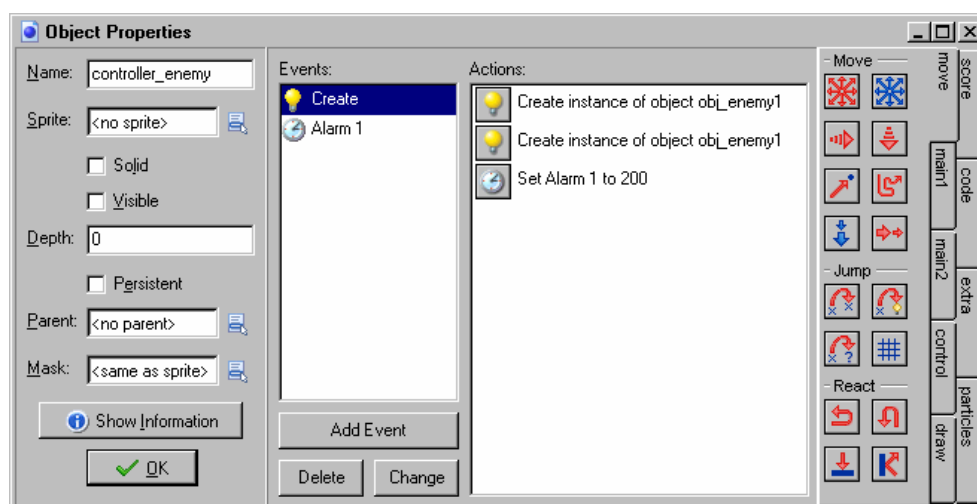
Het verschijnen van de vijand

Tot slot voegen we nog een nieuw object toe dat zal beschrijven hoe de vijandige vliegtuigen op het scherm verschijnen. Bij je eerste test heb je gemerkt dat je slechts één vijandig vliegtuig in het spel hebt: dat is een gevolg van het feit dat er slechts één object in room_sea werd geplaatst. We willen het spel nu aantrekkelijker maken door verschillende vijandige vliegtuigen in het spel te brengen: hoe langer het spel loopt, hoe meer vijandige vliegtuigen moeten opduiken. Daardoor wordt het spel steeds moeilijker.

We creëren hiervoor een nieuw object controller_enemy. Bij de creatie van dit object worden meteen twee vijandige vliegtuigen in het spel gebracht. Bovendien begint een alarmklok af te tellen. Wanneer het alarm afgaat, wordt een nieuw vijandig vliegtuig in het spel gebracht. Op die manier zullen de vijanden steeds toenemen.

object	gebeurtenis (event)	handeling (action)
controller_enemy	van zodra de controller gecreëerd is	er worden twee vijandige vliegtuigen in het spel gebracht, een alarmklok begint af te tellen van 200.
controller_enemy	wanneer het alarm af gaat	er wordt opnieuw een vijandig vliegtuig aangemaakt, het alarm wordt weer op 500 geplaatst.

Op basis van deze tabel krijg je de twee onderstaande events en de bijhorende acties: bij het aanmaken van een instance van obj_enemy1 kies je als x-coördinaat `random(room_width)` en als y-coördinaat `-16`.



Vervang in room_sea nu de obj_enemy1 door controller_enemy en voeg ook controller_life toe, telkens op een willekeurige plaats. Aangezien aan deze objecten geen sprite is toegekend, zal een vraagteken weergegeven worden bovenop de achtergrond.

Het spel is nu helemaal af en klaar om getest te worden.



Sta nu even stil bij de nieuwigheden die je in dit spel hebt leren kennen:

1. Je hebt geleerd dat je variabelen kunt gebruiken om bepaalde commando's toe te laten of niet. Denk hierbij aan de variabele can_shoot.
2. Je hebt kennis gemaakt met objecten die niet verbonden zijn aan een sprite, maar wezenlijk zijn voor het spel: een object om de score bij te houden en een object om het verschijnen van vijandige vliegtuigen te controleren.
3. Je hebt geleerd hoe je een bewegende achtergrond kunt maken en hoe objecten kunnen bewegen met een vaste snelheid.
4. Je hebt geleerd dat na afloop van een gebeurtenis, sprites kunnen vernietigd worden.

Oefening

Je eerste scrolling shooter is helemaal klaar, maar vrij eenvoudig om te spelen. Je kunt de moeilijkheidsgraad verhogen door meer vijandige vliegtuigen te creëren.

Lees de onderstaande aanpassingen aandachtig. Ga daarna systematisch te werk:

- maak een lijst met de sprites en objecten die toegevoegd moeten worden.
- Maak een tabel met per object de events en actions.
- Pas op basis van je werkschema je programma aan.

In deze paragraaf gaan we drie nieuwe types van vijand toevoegen. Eentje dat zelf ook kogels afschiet verticaal en naar beneden, een tweede dat kogels afschiet in de richting van het vliegtuig van de speler, een derde die geen kogels afschiet, maar van de andere kant komt gevlogen. Deze drie vliegtuigen duiken pas later op in het spel.

Om een tweede vijand te creëren, maken we eerst een nieuwe sprite `spr_enemy2` aan. Kies een andere figuur als sprite. We maken ook een nieuw object aan, maar vertrekken van een kopie van `obj_enemy1`. Klik met de rechtermuisknop op `obj_enemy1` en kies **Duplicate**. Dubbelklik op het nieuwe object om de instellingen te veranderen. Geef het de naam `obj_enemy2` en pas de sprite aan. Omdat dit een ander type vijand is, krijgt de speler 10 punten bij het stukschieten van deze vijand. Pas dus de instelling bij het collision-event aan.

Om deze vijand ook te laten schieten, creëren we een nieuwe sprite `spr_bullet2` en een nieuw object `obj_bullet2`. Bij het create-event bij dit object krijgt de sprite een verticale snelheid naar beneden. Bij het step-event zorg je ervoor dat de kogel verdwijnt als die de rand van de pagina heeft bereikt. Bij het collision-event met het vliegtuig van de speler, vermindert de health van de speler met -5, wordt de kogel vernietigd en klinkt er een geluid.

We moeten er nu nog voor zorgen dat het vijandig vliegtuig af en toe kan vuren. We regelen dit via het step-event van het vliegtuig. We gebruiken een nieuwe actie "with a chance perform next action" en nemen als aantal zijden van de dobbelsteen 30. De volgende actie wordt dus om de 30 stappen uitgevoerd. De actie die volgt is het creëren van `obj_bullet2`.

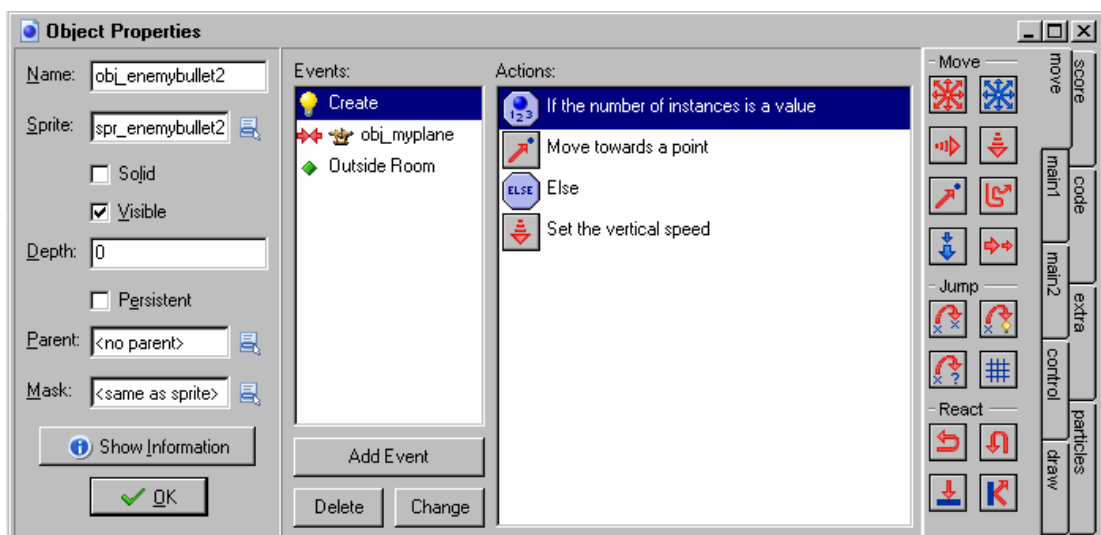
Om er voor te zorgen dat het tweede vijandig vliegtuig ergens opduikt in het spel passen we controller_enemy aan. In het create-event zet je `alarm1` op waarde 1000. Maak nu een `alarm1`-event met de volgende acties: creëer `obj_enemy2`, zet `alarm1` terug op 500 om een bijkomende tweede

vijand te creëren. Het eerste vliegtuig van dit type duikt dus op na 30 seconden, het tweede na 15, enz.

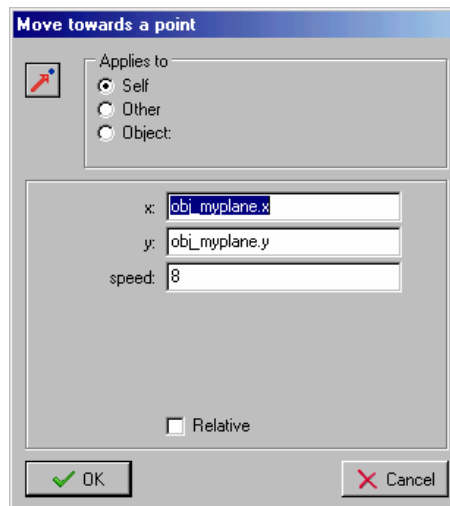
We creëren nu een derde type vijand, dus een nieuwe sprite en een nieuw object. Je kunt een duplicaat maken van `obj_enemy2` en enkele instellingen aanpassen: de punten bij het afschieten verhogen met 20. Kies nu ook een derde sprite voor een nieuwe kogel `spr_bullet3` en een bijhorend object `obj_bullet3`. Deze kogel wordt opnieuw aangemaakt in het step-event van het vliegtuig. Gebruik opnieuw een dobbelsteen, maar kies nu als parameter 80.

We willen dit derde type vijand nu een kogel laten afvuren in de richting van het vliegtuig van de speler. We passen de instellingen van `obj_bullet3` dus verder aan. Bij het create-event nemen we de actie "move towards a point". De vraag is natuurlijk naar welk punt? Omdat we de coördinaat van het vliegtuig van de speler willen aangeven gebruiken we `obj_myplane.x` en `obj_myplane.y`. Dit komt overeen met de x en y-variabele van het object `obj_myplane`. Deze werkwijze zorgt voor een probleem als er tijdens het afvuren van de kogel geen vliegtuig van de speler aanwezig is, omdat het net geëxplodeerd is bijvoorbeeld. Daarom moet er eerst een IF-test uitgevoerd worden, die nagaat of er wel een spelersvliegtuig aanwezig is. Dat doen we door een actie die nagaat hoeveel keer een object in het spel aanwezig is (het aantal instances van één object). Wanneer dit nul is, wordt de kogel verticaal afgevuurd.

Je bekomt dus de volgende objectinstellingen:



Voor de actie "move towards a point", neem je de volgende instellingen:



Omdat het nu wat moeilijker is na te gaan of de kogel buiten het scherm van het spel komt, gebruiken we een nieuw event "OUTSIDE ROOM". De actie die hierbij hoort is "destroy the instance".

Het enige wat we nu nog moeten doen is controller_enemy nogmaals aanpassen. We voegen bij het create-event nog een alarmklok toe, alarm2. Stel dit alarm in op 2000. Maak ook een bijhorend event alarm2: creëer bij het afgaan van dit alarm een nieuwe vijand van dit type en zet het alarm2 weer op 1000.

Daarmee is ook onze derde vijand helemaal klaar. We voegen nu nog de laatste vijand toe. Deze is net hetzelfde als de eerste vijand, alleen de snelheid is in de andere richting. Je zult opnieuw controller_enemy moeten aanpassen en ook een alarm3 definiëren.

Het spel is nu helemaal speelklaar. Het spel zal steeds moeilijker worden. Uiteraard kun je nu je creativiteit de vrije loop laten, achtergrondmuziek toevoegen, nieuwe sounds toevoegen, nog meer vijanden creëren die ook horizontaal bewegen...

Je eigen spel

Je hebt nu genoeg kennis en ervaring om je eerste eigen spel te ontwikkelen. Je krijgt een grote vrijheid in het bedenken van het concept. Enkel de volgende elementen moeten in je spel aanwezig zijn:

- De speler loopt rond in een wereld waar die door op verschillende plaatsen te komen, punten kan verzamelen.
- Op regelmatige tijdstippen moet de speler een glas alcohol drinken: daardoor verhoogt z'n BAC-waarde
- Dit heeft als gevolg dat de speler de zaak steeds moeilijker onder controle houdt.
- De speler wordt achtervolgd door z'n rivalen: wanneer deze hem vernietigen, verliest de speler een leven.
- Maak een scorebord waar de score wordt bijgehouden, het aantal levens en de stand van de BAC-waarde van de speler.
- Op het einde krijgt de speler de kans om z'n naam in de puntentabel in te geven.