Game Maker 6.1

The next level

Hoofdstuk 2.2

The next level

Het wordt tijd om een nieuw level aan ons spel toe te voegen. Letterlijk en figuurlijk.

Je hebt al een begin van een speelbaar spel gemaakt, maar er moet natuurlijk ook een vervolg en een einde aan je spel komen. Het **design document** slaan we hier even over. Aan het einde schrijf je zelf op wat je gemaakt hebt.
Laten we even beginnen met een herhaling van wat je geleerd hebt.

- Open je spel met het stuiterballetje.
- Maak een nieuwe sprite met de naam *sprt_blokje_extra_leven.*
- Maak een nieuw object met deze sprite.
- Maak het object **solid.**

OK, dat moet niet al te moeilijk zijn geweest. We gaan eerst een achtergrondgeluid toevoegen aan je spel.

- Maak een nieuw geluid aan.
- Kies een *.mid-bestand* . Deze muziekbestandjes zijn langer dan de effect geluidjes.
- Noem het geluid *snd_achtergrondmuziek.*
- Ga naar het object *obj_controler.*
- Voeg aan het event  *Game start* de action  *sound* toe, en kies hiervoor je muziekje.

- Ga naar *rm_level1.*
- Maak in *rm_level1* een patroon dat alleen bestaat uit *obj_blokje_enkel.*

Klik nog een keer met de linker muisknop op *rm_level1.*
Kies voor *Duplicate.*

Je hebt twee rooms.

- Noem de tweede room rm_*level2.*
- Herhaal dit tot je vier levels hebt.
- Maak voldoende sprites en objecten aan voor vier blokjes. Maak ze **solid**.
- Maak een geluid aan dat afspeelt aan het einde van ieder level. Dus add  sound. Noem dit geluid *succes.*
- Maak voor ieder level een nieuwe achtergrond.

Laten we eens tellen wat we allemaal hebben.

7 sprites die beginnen met *sprt_*
5 geluiden die beginnen met *snd_*
4 achtergronden die beginnen met *back_*
8 objecten die beginnen met *obj_*
4 levels die beginnen met *rm_,* let er bij de rooms op dat de settings bij alle rooms gelijk zijn !!!

Als je dit allemaal hebt kunnen we de *events* en *actions* aan de objecten te gaan voegen.

 Sla je spel op.

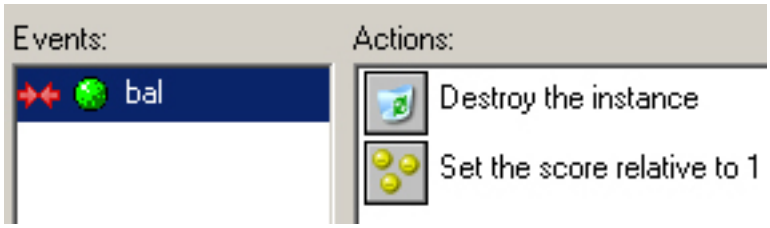
Voor we aan de opbouw van de levels beginnen maak je nog één nieuw blokje aan. Dit blokje heeft straks de eigenschap dat het niet verdwijnt als de bal er tegen aan komt.

- Maak de sprite.
- Maak het object en noem dit *obj_blokje_solid.*

Nu wordt het tijd om aan alle blokjes de juiste eigenschappen te gaan geven.

Zoals aan alle objecten moeten ook aan *obj_blokje3* en *obj_blokje4* nog events en actions toegevoegd worden, maar je moet ook gaan kijken of de eigenschappen van *obj_blokje_enkel* en *obj_blokje_dubbel* nog van toepassing zijn in dit nieuwe spel.

- Dubbelklik op *obj_blokje_enkel*.



Je ziet dat *obj_blokje_enkel* verdwijnt als het door de bal geraakt wordt en de speler krijgt er 1 punt bij. Dit is voor mijn spel goed, maar je mag het natuurlijk aanpassen.

- Dubbelklik op *obj_blokje_dubbel*.

Hier zie je dat *obj_blokje_dubbel* veranderd in *obj_blokje_enkel* als het geraakt wordt. De speler krijgt hier nog geen punten voor. Dat gaan we veranderen.

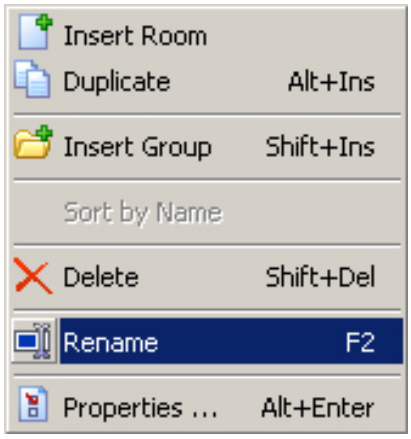
- Voeg een score toe van 2 punten per geraakt blokje. Denk er goed aan dat deze score *relative* moet zijn.

 Sla je spel op.

Wat zou je willen dat er met *obj_blokje3* gebeurt als de bal het raakt? En wat gebeurt er met *obj_blokje4*? Ik laat je hier één mogelijkheid zien, maar je kunt natuurlijk ook iets anders verzinnen.

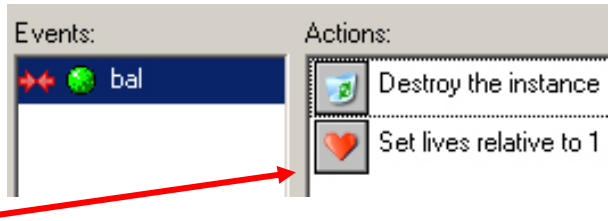
Bedenk eerst heel goed wat het derde blokje moet gaan doen. Ik wil dat de speler een nieuw leven krijgt.

- Klik met je rechtermuisknop op *sprt_blokje3.*
- Kies voor *rename.*
- Noem de sprite *sprt_blokje_extra_leven.* Let er op de je geen spaties gebruikt, maar _ .
- Doe nu hetzelfde met het object van blokje3.
- Dubbelklik op *obj_blokje_extra_leven.*



Ik wil graag dat *obj_blokje_extra_leven* verdwijnt als het door de bal geraakt wordt. Op het moment dat *obj_blokje_extra_leven* verdwijnt krijgt de speler er een leven bij.

- Add event *collision with bal.*
- Kies action *destroy the instant*
- Kies action *set live relative to 1.*



Vergeet niet dat de levens *relative* zijn, anders zet je de totale levensstand op 1.

Nu verder met blokje4 rename het blokje als *sprt_blokje_event.* Pas ook het object van blokje4 aan. Waar je nu het woord *event* ziet staan vul je in wat er met het blokje gebeurt als het door de bal geraakt wordt.

- Dubbelklik op *obj_blokje_event.*
- Kies een event en een of meerdere actions.
- Als je zelf niets kunt bedenken vind je verderop dit boekje een aantal mogelijkheden.

We moeten *obj_blokje_solid* nog een functie geven.

- Dubbelklik op *obj_blokje_solid.*
- Vink **solid** aan.

- Dubbelklik op obj_*bal.*

- Add event  *collision with obj_blokje_solid.*
- Kies action  *bounce against solid object.*
- Herhaal dit voor *obj_blokje_extra_leven* en *obj_blokje_verander_bal*.

 Sla je spel op.

We beginnen met het meest simpele. Het opbouwen van de levels. *Level1* bestaat helemaal uit *obj_blokje_enkel*-objecten.

- Kies voor ieder level een andere achtergrond.
- Bouw level2 op uit *obj_blokje_enkel en obj_blokje_dubbel* objecten.
- Bouw level3 op uit *obj_blokje_enkel*, *obj_blokje_dubbel*, *obj_blokje_extra_leven* en *obj_blokje_solid* objecten.
- Bouw level4 op uit *obj_blokje_enkel*, *obj_blokje_dubbel*, *obj_blokje_event* en *blokje_solid* objecten.

 Sla je spel op.

Nu de levels klaar zijn moeten we ook van het eerste naar het tweede level kunnen gaan.

Om van het ene level naar het andere te gaan moeten we ***variables*** (variabelen) instellen. Een variable voert een actie uit als aan een bepaalde voorwaarde voldaan wordt. De variable die we hier gaan gebruiken kan '***true***' (ja) of '***false***' (nee).
Bij *level1* willen we graag dat het spel naar het volgende level gaat als alle *obj_blokje_enkel*-objecten weg zijn. De variable is dus *obj_blokje_enkel.*

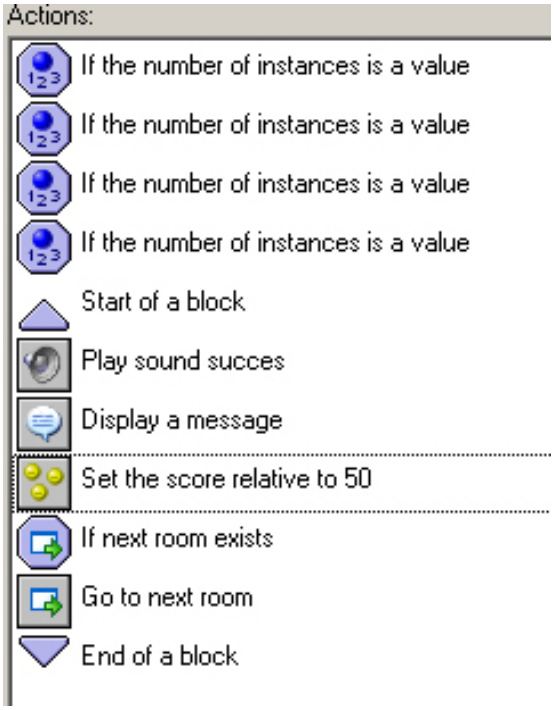
- Add event  *step*
- Kies action  *If the number of an instance is a value.* Uit het tabblad **control**.
- De instance is in dit geval *obj_blokje_enkel*.

Je moet beschrijven wat er gebeurt als het level uit is. Eerst beschrijf je dat met woorden, als het level uit is:

1. Hoort de speler een muziekje *succes*.
2. Krijgt hij 50 punten voor het uitspelen van het level.
3. Krijgt hij een bericht dat het level uit is.
4. Gaat hij naar het volgende level.

Je wilt dat deze actions alleen uitgevoerd worden als alle blokjes weg zijn, daarom moet je deze actions koppelen aan de variable. Dat doe je op de volgende manier.

- Maak het geluidje voor *snd_succes.*
- Kies action  uit tabblad **contol**.
- Kies action  en kies voor het geluidje *succes.*
- Kies action  *set the score relative to 50.*
- Kies action  en typ een berichtje dat je aan het einde van ieder level wilt laten zien.
- Kies action  *if next room exists* (als de volgende room bestaat) uit tabblad **main1**.



- Kies action  *go to next room*, kies je eigen overgang.
- Kies action 

Je gaat van *level1* naar *level2* op het moment dat alle blokjes in *level1* weg zijn. Hetzelfde wil je natuurlijk doen met *level2* en *level3*. Dat is heel eenvoudig. Voeg gewoon nog 3 variables toe aan *step.*

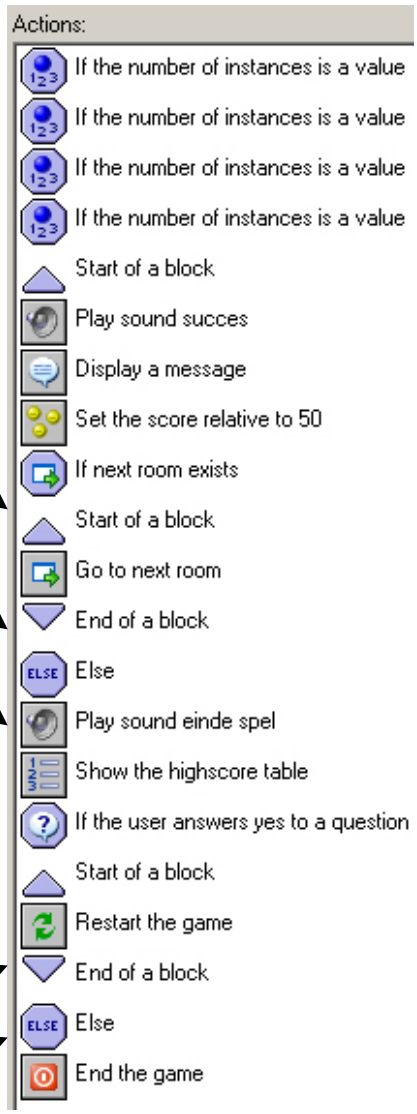
- Kies action  *If the number of an instance is a value.* Uit het tabblad **control**.
- De instance is in dit geval *obj_blokje_dubbel.*
- Doe hetzelfde voor *obj_blokje_extra_leven* en *obj_blokje_event*

Maar na *level4*, is geen level meer. Het spel moet daar dus stoppen. Schrijf weer op wat er moet gebeuren op het moment dat het spel eindigd.

1. Er moet een geluidje gespeeld worden.
2. De highscoretabel moet in beeld komen.
3. De speler moet de keuze krijgen om opnieuw te starten.
4. Als je speler voor 'yes' kiest wordt het spel opnieuw gestart.
5. Als de speler voor 'no' kiest wordt het spel afgesloten.

In deze omschrijving zitten heel veel variables. De eerste variable is de vraag of er een volgende room is. Je had bij level1 t/m level3 bepaald dat *If next room excists* (als er een volgende room is) het spel naar de volgende room moest gaan.

Nu is het antwoord op de vraag 'nee' of ***false***. Er is geen volgende room meer. Er moet dus een nieuwe action gemaakt worden voor ***false***.



- Plaats *Go to next room* tussen *blocks*.  .
- Kies action  , deze komt in actie als de variable ***false*** is.
- Kies actions  *sound* en  *highscore*.
- Kies action  en type de vraag. 'Wil je het spel opnieuw starten?'

Ook hier is weer een variable. Het antwoord op de vraag kan zowel ja als nee zijn.

- Plaats de action  dus weer tussen blocks.
- Maak ook weer een action  aan.
- Als laatste maak je een action *End game* uit tabblad **main2**.

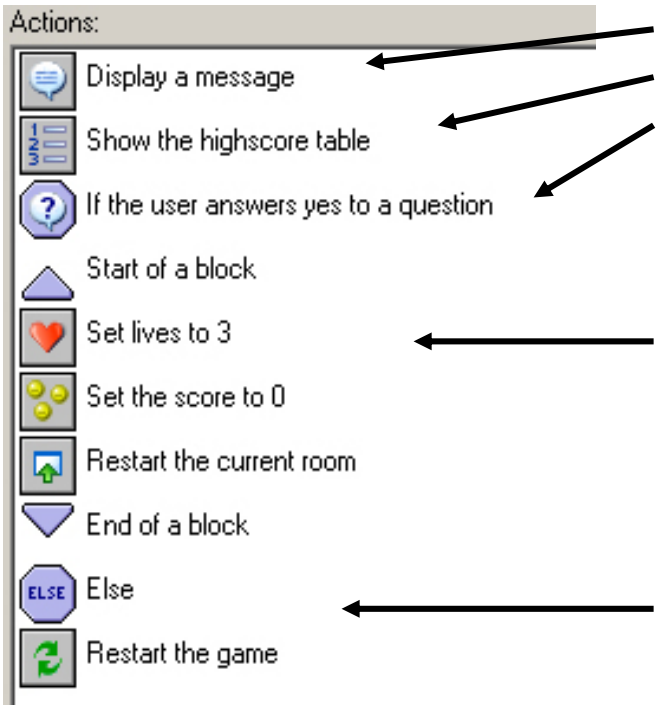
Game over continued

Je hebt al een aantal functie gemaakt voor het geval de speler al zijn levens kwijt is vóór hij het einde van het spel bereikt. Daar gaan we nog iets aan toevoegen. Als de speler bijvoorbeeld in *rm_level2* af zou gaan, kan hij er voor kiezen weer in *rm_level2* te starten en niet naar het begin van het spel te gaan.

We gaan het event *No more lives* helemaal opnieuw schrijven.

- Dubbelklik op *bal* en verwijder het event *No more lives*
- Dubbelklik op *obj_controler* en maak een nieuw event *No more lives.*

Beschrijf eerst weer in een aantal stappen wat je wil dat er gebeurt. Denk er goed aan dat de volgorde hierbij erg belangrijk is.



1. Melding '*game over*'
2. Laat de highscore zien
3. Vraag de speler of hij op het laatste level wil starten. *'Wil je op het laatst gespeelde level starten?'*
4. Als de speler met 'yes' antwoord herlaad je het laatst gespeelde level, de score moet dan weer op 0 beginnen en de speler krijgt weer 3 levens.
5. Als de speler met 'no' antwoord wordt het spel opnieuw gestart bij level1.

Het volledige event ziet er dan zo uit. Je ziet dat hier ook weer gebruik gemaakt wordt van een *block.* Dat komt omdat er een variable is. De speler kan namelijk 'ja' en 'nee' zeggen. Als de speler met 'ja' antwoord wordt het block uitgevoerd. Als de speler met 'nee' antwoord, wordt het block overgeslagen.

Dan komt de action 

(anders) en wordt het spel opnieuw gestart.

In het spel zie je de volgende melding.

 Sla je spel op.

Bonussen

Even terug naar de blokjes. Je hebt al heel wat programmeer-ervaring opgedaan en ik zal je uitleggen hoe je een bonus aan een blokje kunt verbinden. Ik heb er voor gekozen om *obj_blokje_event* in een bonus te veranderen. De bonus valt naar beneden. De bonus krijgt de speler echter niet zomaar. Hij/zij moet de bonus die naar beneden valt opvangen met het schuifje.

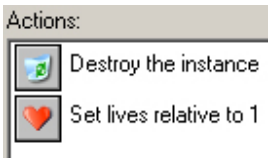
- Rename *sprt_blokje_event* in *sprt_blokje_bonus.*
- Pas ook het object aan.
- Verwijder alle events.

Laten we even op een rijtje zetten wat we moeten programmeren.

1. *Obj_blokje_bonus* moet veranderen in een *obj_bonus*
2. Het *obj_bonus* moet naar beneden vallen.
3. Als de speler de *obj_bonus* met het schuifje opvangt, moet de *obj_bonus* verdwijnen en moet de speler iets krijgen. Positieve, maar ook negatieve bonussen mogen, zoals minpunten, of verlies van levens.

- Maak een sprite aan die de bonus moet voorstellen. Noem deze *sprt_bonus_extra_leven.*
- Maak van deze sprite een object.
- Ga naar *obj_blokje_bonus* en kies bij het event  *collision with bal*
- Kies voor de action  *change instance into obj_bonus_extra_leven.*
- Laat het blokje verdwijnen
- Ga naar *obj_bonus_extra_leven* en add event  *step.*

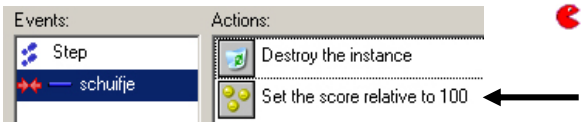
- Kies bij step voor de action  *move* en kies alleen het pijltje naar beneden, speed 3.
- Add event  *collision with schuifje.*



- Laat de *obj_bonus_extra_leven* verdwijnen en de speler er een leven bij krijgen. (zie *obj_blokje_extra_leven)*

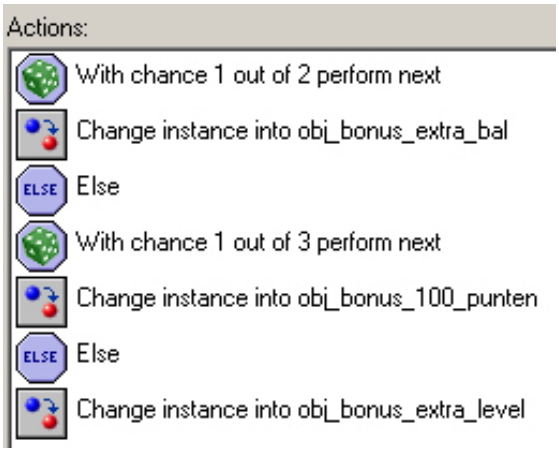
Nog leuker wordt het als er verschillende soorten bonussen naar beneden vallen. Je kunt meerdere bonus-objecten aan maken. Als je deze bonussen *at random* (in een willekeurige volgorde) naar beneden wilt laten vallen kom je al in een ingewikkelder stukje programmeer werk. Wil je het toch proberen, doe dan het volgende.

- Maak voldoende sprites aan voor alle bonussen.
- Dupliceer object *obj_bonus_extra_leven* zodat je voldoende bonussen hebt.
- Bedenk welke bonussen de speler kan krijgen.
- Geef ieder object de juiste sprite en de juiste naam.



- Geef iedere *bonus* zijn eigen eigenschap als hij het schuifje raakt.

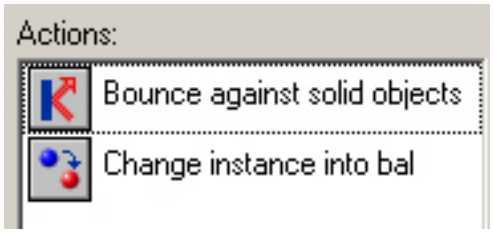
- Programmeer het volgende bij *obj_blokje_bonus*, maak daarbij gebruik van  uit tabblad **control**.
- Geef alle bonussen een 'kans' van 1 *out of* x. Vul voor x een getal in.
- Eindig altijd met  en  zonder  .



 Sla je spel op.

De blokjes hebben allemaal een functie. Ik wil ook nog iets doen met de bal. Ik wil dat de bal van object veranderd als hij *obj_blokje_bonus* raakt.

- Maak een nieuwe sprite met de naam *sprt_bal_kleur.*
- Dupliceer het object van *bal* en noem dit *obj_bal_kleur.*
- Geef *obj_bal_kleur* de juiste sprite.
- Laat *obj_bal* in *obj_bal_kleur* veranderen als hij *obj_blokje_bonus* raakt.



- Laat *obj_bal_kleur* weer in *obj_bal* veranderen als hij *obj_blokje_solid* of de *obj_muur* raakt.

Als je het spel gaat spelen merk je dat de nieuwe bal geen blokjes kan laten verdwijnen.

Probeer eens te verklaren waarom ze geen blokjes laten verdwijnen.

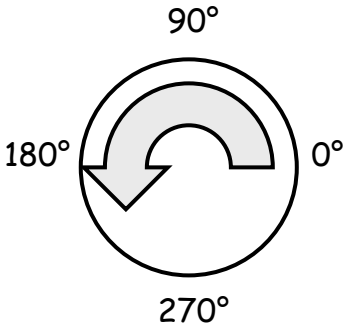
De verklaring is vrij simpel. De blokjes hebben alleen de instructie gekregen om iets te doen als ze in aanraking komen met *obj_bal.* Dit moet je dus nog aanpassen. Dat kan op twee manieren. Of de nieuwe bal een event geven voor een collision met de blokjes, of je blokjes een event geven voor een collision met de nieuwe bal.

- Laat de blokjes ook door een collision met *obj_bal_kleur* verdwijnen.
- Geef de speler 2x zoveel punten voor deze events als bij *obj_bal.*

Ik wil nog iets gaan doen met het balletje. Ik wil graag dat het balletje in een willekeurige richting begint.

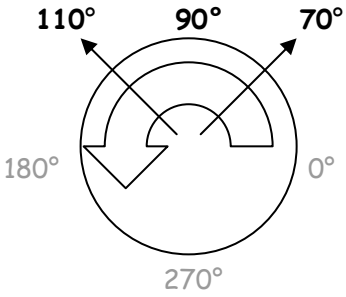
- Ga naar *obj_bal.*
- Kies event  *create.*
- Verwijder de action  move en vervang deze door  *Set direction and speed of motion.*

Hiermee kun je een precieze beweging aangeven. De beweging start dan onder een hoek tussen 0 en 360 graden. 0 betekent naar → rechts. De richting is tegen de wijzers van de klok in.
Bijvoorbeeld, 90 is een beweging recht omhoog.



Wanneer je een willekeurige richting wilt, type je *random(360).* Random betekend willekeurig.

- Type *Direction = 70+random(40)*



Dit betekend dus een beweging van 70 graden + een extra 0 tot 40 graden. Voor de rekenwonders onder jullie, de bal kan dus tussen een hoek van 70 en een hoek van 110 graden starten. (zie tekening). De bal kan dus links én rechts omhoog gaan

- Klik op  OK

De bal begint nu in een willekeurige richting naar boven.

- Schrijf nu achteraf een duidelijk **design document** van het spel dat je gemaakt hebt.
- Maak de  *game information* onder de F1-toets.
- Maak van je spel een executable.

 Sla je spel op.

Vragen

1. Wat is een sprite?

2. Wat betekend *solid?*
   A. doorzichtig
   B. zwaar
   C. massief
   D. onzichtbaar

3. Je ziet deze melding in het spel. Welke action heb je gebruikt?
   A. 
   B. 
   C. 

   D. 

4. Met welke action kun je de score met sprites in beeld laten zien?
   A. 
   B. 
   C. 
   D. 

5. De *direction* van een object is 80+ramdom(20). In welke richting kan dit object gaan bewegen?
    A. 60 graden
    B. 90 graden
    C. 120 graden
    D. 360 graden

