2D Game

beginnen met programmeren

Auteur: Gabor de Mooij
Versie: 1.1 – Januari 2025

Introductie

In deze les maken leerlingen samen met de leraar een echt 2D-computerspel dat zich afspeelt in hun eigen buurt. Hiervoor gebruiken we een Nederlandstalige programmeertaal genaamd Citrine/NL. De leerlingen leren spelenderwijs objectgeoriënteerd programmeren in pure vorm (Smalltalk-stijl). In tegenstelling tot andere platforms en systemen zijn de leerlingen eigenaar van het door hen gemaakte werk. Ze hoeven zich niet te registreren en zijn niet afhankelijk van een externe partij of internet (offline). De leerlingen kunnen op eigen kracht doorgroeien in de programmeertaal en complexe programma's of games maken, zonder beperkingen. Gemaakte werken kunnen eventueel worden overgezet naar een smartphone, gameconsole of website naar keuze!

Duur	90 minuten
Lessen	1 of 2
Voorbereiding	60 minuten (eenmalig) – daarna +/- 15 minuten
Taal	Nederlands
Lengte	30 regels code
Leeftijd	12-16 jaar
Voorkennis	Leerling: - Algemeen computergebruik (aan/uit zetten, bestandsbeheer, kladblok) Leraar: - Basiskennis OOP is een pre maar niet vereist
Leerstof	- Programmeren van een spel in Citrine/NL (Nederlandstalig) - Variabelen - Objectgeorienteerd programmeren (OOP) - X/Y-coordinaten - Werken met conditionele code - Basiskennis kunstmatige intelligentie (AI)
Activiteit	- Klassikaal een opzetje van een spel maken - Leerlingen kunnen spel verder uitwerken bij stap 12 (in groepjes) - Optioneel: huiswerk spel verfijnen/uitbreiden
Nodig	- 1 computer/laptop aangesloten op projector of digitaal schoolbord (leraar) - Diverse (oude) computers voor groepjes leerlingen - Windows, Linux of Mac

	- Kladblok of vergelijkbaar computerprogramma - Citrine/NL (www.citrine-lang.org/nl.ctr)
Niet nodig	- Internet - Het is niet nodig om te registeren
Thema	De les kan ingepast worden in een thema zoals halloween, kerst, pasen, zomer, geschiedenis etc. Pas simpelweg de plaatjes in het spel aan. Geef als inleiding een themaverhaal.
Vakken	Deze les kan gegeven worden in samenwerking met de vakken: - Wiskunde (logica, X-Y coordinaten) - Geschiedenis (thema) - Kunst (ontwerp van plaatjes en afbeeldingen)
Platform	Citrine (platform agnostisch)

Inspiratie

Voordat de les begint kun je leerlingen laten zien wat je kunt maken met de programmeertaal Citrine/NL. Laat ze bijvoorbeeld het demospel Piccolo zien:
https://citrine-lang.org/download.ctr#games

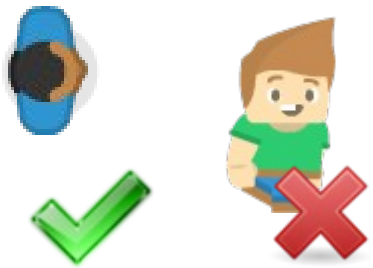
Gamification

Gedurende de les kan het zijn dat je als leraar per ongeluk een foutje maakt. Spreek vantevoren met je leerlingen af dat ze debugpunten kunnen verdienen door foutjes vooraf te spotten of als er een foutje is ingeslopen, deze op te sporen.

Lespakket

Voorbereiding (leraar)

- Installeer Citrine op de gewenste computer
- Maak een schermafdruk via Google Maps van je schoolplein of een grasveld in de omgeving
- Download plaatjes voor spelers, tegenstanders en obstakels op bijvoorbeeld kenney.nl (https://kenney.nl/assets/top-down-shooter, https://kenney.nl/assets/sports-pack, https://kenney.nl/assets/sokoban) of https://opengameart.org/
- Je mag de leerlingen natuurlijk ook zelf plaatjes laten tekenen met een tekenprogramma
- Let erop dat de plaatjes van 'bovenaf' zijn en dat de neus naar rechts wijst, zoniet pas het plaatje dan aan in een tekenprogramma. Maak alle plaatjes ongeveer 40 x 40.



- Je hebt de volgende plaatjes nodig:

1. Achtergrond (van bovenaf) – bijvoorbeeld schoolplein, voetbalveld, grasveld (egaal, liever geen obstakels of muren in het zicht), gewoon groen/grijs mag ook. Noem de achtergrond: "achtergrond.png"
2. Een plaatje van de speler van bovenaf met de neus naar rechts. Noem de speler: "speler.png".
3. Een plaatje van de tegenstander van bovenaf met de neus naar rechts. Noem de tegenstander: "boef.png".
4. Een plaatje van een obstakel bijvoorbeeld een kist, boom of muur. Noem deze "obstakel.png".
5. Een plaatje van een schat, ster of ander doel "doel.png".

Test het volledige programma, zoals getoond aan het einde van dit document ter voorbereiding. Bij elke stap wordt een werkend deelprogramma getoond.

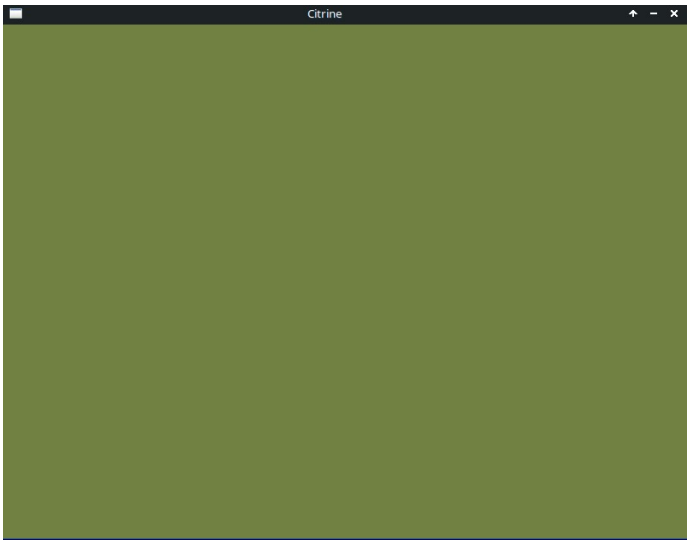
Stap 1 – De achtergrond tonen op het scherm

- Open kladblok
- Schrijf een regel code:

```
Media scherm: ['achtergrond.png'].
```

- Sla het bestand op, noem het “spel.ctr”.
- Sleep het bestand naar het pictogram van Citrine
- Je ziet nu jullie schoolplein / speelveld van bovenaf

In het voorbeeld gebruiken we gewoon een groene achtergrond.



Veelvoorkomende fouten:
- Achtergrond verschijnt niet: klopt de bestandsnaam wel?
- Programma start niet: ben je een punt vergeten?
- Als je er niet uitkomt, start het programma dan vanaf de opdrachtregel:
ctrnl spel.ctr
Als er een fout optreedt zie je een melding verschijnen met een regelnummer

Leg de code uit:

- Wat gebeurt er? De computer zit vol met objecten. Elk object kan dingen doen. Om een object ‘iets’ te laten doen stuur je een berichtje. Wij sturen het bericht ‘scherm’ naar het object Media. Je mag ook een extraatje sturen, dat noemen we een argument. In dit geval sturen het plaatje dat op het scherm moet komen ‘achtergrond.png’. Het Media-object doet alles wat te maken heeft met plaatjes, geluidjes, muziekjes en de gamepad of joystick.
- Tijd voor vragen

Stap 2 – Variabelen

Je kunt een object ook een bijnaam geven, de computer onthoudt de bijnaam van het object. Als je dan de naam opschrijft weet de computer welk object je bedoelt.

```
>> m := Media nieuw.
m scherm: ['achtergrond.png'].
```

In dit voorbeeld sturen we het bericht ‘nieuw’ naar Media. Er zit deze keer geen extraatje bij. Door het bericht ‘nieuw’ te sturen krijgen we een eigen versie van het Media-object. We hoeven dan ook niet de hele tijd meer ‘Media’ te typen, we kunnen nu gewoon ‘m’ zeggen want zo hebben we ons eigen Media-object genoemd. Met de pijltjes >> reserveren we een plekje in het geheugen van de computer om dit object op te slaan. De computer gaat dan uitrekenen hoeveel plek er nodig is.
Met := geven we een object een naam.

Veelvoorkomende fouten:
- Gebruik je de juiste aanhalingstekens? Geen punt vergeten?

Stap 3 – Plaatje

We gaan nu een plaatje op de achtergrond zetten (laat het plaatje zien).
Pas de code aan:

```
>> m := Media nieuw.
>> doel := Plaatje nieuw: ['doel.png'].
m scherm: ['achtergrond.png'].
```

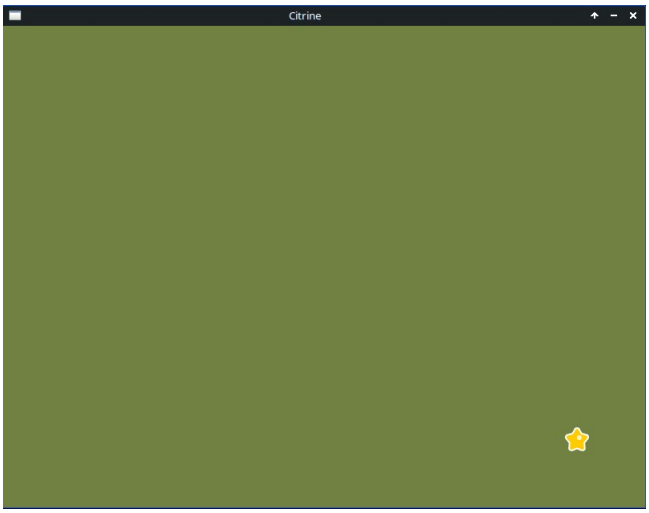
In het voorbeeld is doel een ster:

- Vraag aan de leerlingen wat hier gebeurt
- Laat het resultaat zien

Stap 4 – Coördinaten

- Vraag aan de leerlingen hoe ze het doel een andere plek op het scherm kunnen geven.
- Leg uit dat we links-rechts aanduiden met X, en boven-onder met Y.

```
>> m := Media nieuw.
>> doel := Plaatje nieuw: ['doel.png'].
doel x: 700 y: 500.
m scherm: ['achtergrond.png'].
```



- Laat de leerlingen verschillende waarden invullen voor X en Y.

Stap 5 – Komma

- Als je meerdere keren een bericht wilt sturen mag je een komma gebruiken:

```
>> doel := Plaatje nieuw: ['doel.png'].
doel x: 700 y: 500.
```

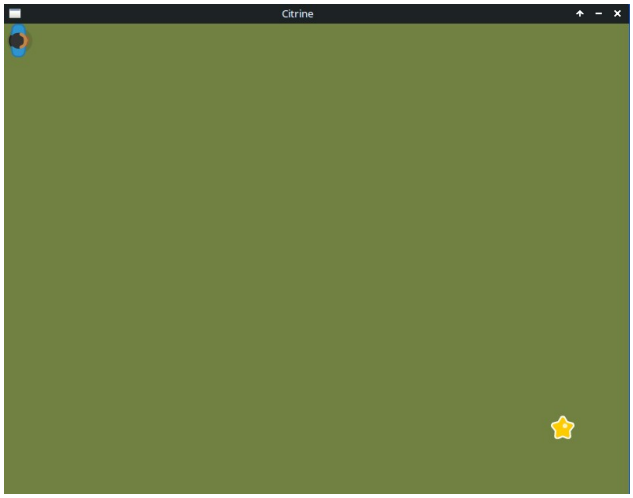
Wordt:

```
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
```

Stap 6 – Joystick

- Sluit een joystick of gamepad aan
- Pas de code aan:

```
>> m := Media nieuw.
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1.
m scherm: ['achtergrond.png'].
```



Laat zien dat de speler nu bestuurt kan worden via de joystick.
Laat de leerlingen de joystick even vasthouden.

Stap 7 – Condities

- We gaan nu zorgen dat wie het doel bereikt wint.
- We sturen het bericht ‘actief: Ja’ naar speler
- Hierdoor weet de computer dat er dingen met de speler kunnen gebeuren
- Een speler kan nu bijvoorbeeld botsen met een ander plaatje
- Om te zorgen dat er bij een botsing iets gebeurt maken we een taak
- Een taak is een reeks opdrachten tussen { en }
- We kunnen de taak koppelen aan de gebeurtenis met het bericht ‘bij: X doen: Taak’
- In dit geval willen we dat er iets gebeurt als we botsen met het doel, dus schrijven we:

```
bij: [‘bots’] doen:
```

- We moeten natuurlijk wel weten waar we tegenaan botsen…! Gelukkig vertelt de computer ons dat aan het begin van de taak. We mogen dit object een eigen naam geven, we kiezen voor de naam ‘ander’. We controleren nu of het doel is bereikt met:

```
ander = doel
```

- Als dat zo is, voeren we een taak uit, dus zetten we de code weer tussen { … }
- In dat geval sturen we het bericht toon: naar Media met als argument ‘Gewonnen!’
- Daarna laten we het programma stoppen door het bericht ‘einde’ te sturen

- Pas de code aan:

```
>> m := Media nieuw.
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1, actief: Ja,
bij: ['bots:'] doen: { :ander
      ( ander = doel ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
      }.
}.
m scherm: ['achtergrond.png'].
```

- Laat de leerlingen even het spel spelen
- Bespreek in de klas dat dit wel erg simpel is en dat het leuker zou zijn als er ook boeven zijn die je achterna gaan zitten

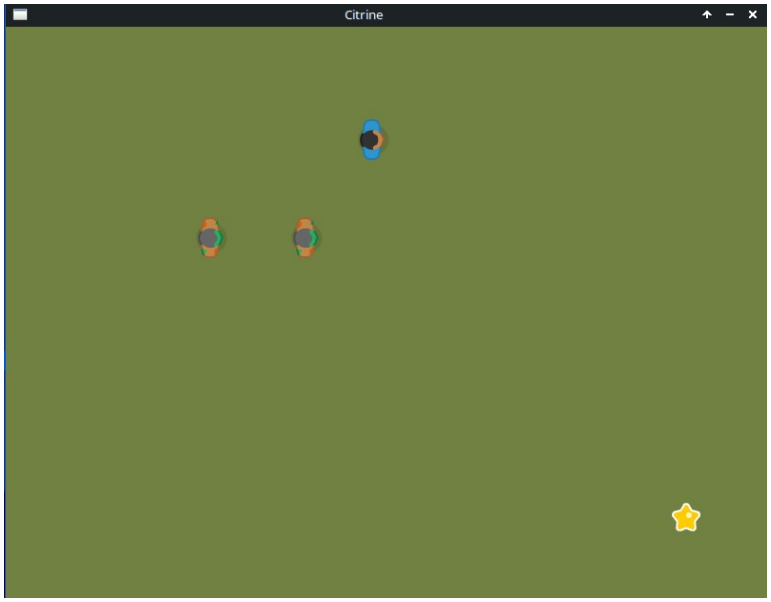
Stap 8 – Tegenstanders

- Zet twee boeven neer op de achtergrond
- Laat de leerlingen bepalen op welke posities

```
>> boef1 := Plaatje nieuw: ['boef.png'], x: 200 y: 200.
>> boef2 := Plaatje nieuw: ['boef.png'], x: 300 y: 200.
```

Voorbeeld:

```
>> m := Media nieuw.
>> boef1 := Plaatje nieuw: ['boef.png'], x: 200 y: 200.
>> boef2 := Plaatje nieuw: ['boef.png'], x: 300 y: 200.
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1, actief: Ja,
bij: ['bots:'] doen: { :ander
      ( ander = doel ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
      }.
}.
m scherm: ['achtergrond.png'].
```



- Laat het programma draaien
- Vraag aan de leerlingen hoe het spel leuker gemaakt kan worden:
    - Er gebeurt niks als je een boef aanraakt
- De boeven bewegen niet

Stap 9 – Invuloefening

- Wat moet er op de puntjes komen?

```
bij: ['bots:'] doen: { :ander
      ( ander = ... of: ander = ... ) ...: {
            Programma einde.
      }.
```

Volledige code:

```
>> m := Media nieuw.
>> boef1 := Plaatje nieuw: ['boef.png'], x: 200 y: 200.
>> boef2 := Plaatje nieuw: ['boef.png'], x: 300 y: 200.
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1, actief: Ja,
bij: ['bots:'] doen: { :ander
      ( ander = boef1 of: ander = boef2 ) ja: {
            Programma einde.
      }.
      ( ander = doel ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
      }.
}.
m scherm: ['achtergrond.png'].
```

Stap 10 – Beweging

Zet boef 1 op 20,600 en boef 2 op 750,600.
Stuur het bericht ‘snelheid: 2’.
Laat de boeven naar de speler lopen:

```
boef1 naar-x: 20 y: 0.
boef2 naar-x: 750 y: 0.
```

Volledige code:

```
>> m := Media nieuw.
>> boef1 := Plaatje nieuw: ['boef.png'], x: 20 y: 600, snelheid: 2.
>> boef2 := Plaatje nieuw: ['boef .png'], x: 750 y: 600, snelheid: 2.
>> doel := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1, actief: Ja,
bij: ['bots:'] doen: { :ander
      ( ander = boef1 of: ander = boef2 ) ja: {
            Programma einde.
      }.
      ( ander = doel ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
      }.
}.

boef1 naar-x: 20 y: 0.
boef2 naar-x: 750 y: 0.
m scherm: ['achtergrond.png'].
```

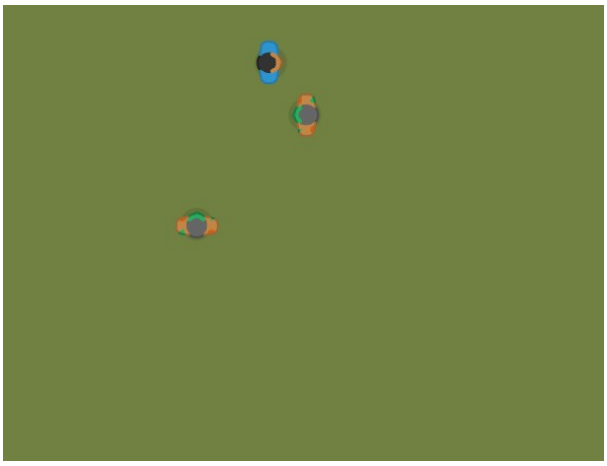
- ***Optioneel*** Beide boeven veranderen wel van Y maar niet van X positie. Hun Y positie blijft dus steeds hetzelfde, leg uit dat je dus ook de oude Y positie kunt doorgeven:

```
boef1 naar-x: 20 y: boef y?.
boef2 naar-x: 750 y: boef y?.
```

- Je kunt de Y-positie opvragen door het bericht y? Te sturen. Voor X is dat x?

Stap 11 – AI

Om het spel echt leuk te maken moeten de boeven intelligenter worden. Ze moeten weten waar de speler zit.



De boeven maken een plan van aanpak. De ene gaat links staan en de ander rechts. Er worden twee wekkers gezet. Als de ene wekker gaat dan loopt de ene boef horizontaal naar de speler en de ander verticaal. Als de andere wekker gaat wisselen ze om. Zo hopen ze de speler te omsingelen.

Laat de leerlingen de volgende code voor de AI invullen:

```
m bij: ['wekker:'] doen: { :nummer
      (nummer = 1) ja: {
            boef1 naar-x: ... x? y: boef1 y?.
            boef2 naar-x: boef2 x? y: ... y?.
            m wekker: 2 over: 1000.
      }.
      (nummer = 2) ja: {
            boef1 naar-x: ... x? y: speler y?.
            boef2 naar-x: speler ... y: ... y?.
            m wekker: ... over: 1000.
      }.
}.
```

Volledige code:

```
>> m := Media nieuw.
>> boef1 := Plaatje nieuw: ['boef.png'], x: 20 y: 600, snelheid: 2.
>> boef2 := Plaatje nieuw: ['boef.png'], x: 750 y: 600, snelheid: 2.
>> schat := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1.
speler actief: Ja, bij: ['bots:'] doen: { :ander
      ( ander = boef1 of: ander = boef2 ) ja: {
            Programma einde.
      }.
      ( ander = schat ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
```

```
        }.
}.
m bij: ['wekker:'] doen: { :nummer
        (nummer = 1) ja: {
                boef1 naar-x: speler x? y: boef1 y?.
                boef2 naar-x: boef2 x? y: speler y?.
                m wekker: 2 over: 1000.
        }.
        (nummer = 2) ja: {
                boef1 naar-x: boef1 x? y: speler y?.
                boef2 naar-x: speler x? y: boef2 y?.
                m wekker: 1 over: 1000.
        }.
}.
m wekker: 1 over: 0, scherm: ['achtergrond.png'].
```

- Laat de leerlingen het spel spelen, het is nu serieus moeilijk!

Stap 12 – Obstakels

- Laat de leerlingen 3 obstakels neerzetten waarachter de speler zich kan verschuilen, door deze slim neer te zetten en er slim mee om te gaan in het spel kun je de boeven ontwijken (maar het is nog steeds behoorlijk moeilijk!)

Om een obstakel neer te zetten moet je een plaatje maken:

```
>> ding1 := Plaatje nieuw: ['obstakel.png']
```

en dan een bericht ‘muur’ sturen met als argument Ja.
Verder moet je nog X en Y zetten.

i.e.

```
>> ding1 := Plaatje nieuw: ['obstakel.png'], muur: Ja, x: ... y: ....
```

Laat de leerlingen dit zelf doen in hun eigen versie. Dit kan individueel of in groepjes.
Als ze het leuk vinden mogen ze ook muziek en geluidjes toevoegen of het spel veranderen.

Volledige code:

```
>> m := Media nieuw.
>> boef1 := Plaatje nieuw: ['boef.png'], x: 20 y: 600, snelheid: 2.
>> boef2 := Plaatje nieuw: ['boef.png'], x: 750 y: 600, snelheid: 2.
>> ding1 := Plaatje nieuw: ['obstakel.png'], muur: Ja, x: 200 y: 200.
>> ding2 := Plaatje nieuw: ['obstakel.png'], muur: Ja, x: 400 y: 300.
>> ding3 := Plaatje nieuw: ['obstakel.png'], muur: Ja, x: 700 y: 400.
>> schat := Plaatje nieuw: ['doel.png'], x: 700 y: 500.
>> speler := Plaatje nieuw: ['speler.png'], besturing: 1.
speler actief: Ja, bij: ['bots:'] doen: { :ander
      ( ander = boef1 of: ander = boef2 ) ja: {
            Programma einde.
      }.
      ( ander = schat ) ja: {
            m toon: ['Gewonnen!'].
            Programma einde.
      }.
}.
m bij: ['wekker:'] doen: { :nummer
      (nummer = 1) ja: {
            boef1 naar-x: speler x? y: boef1 y?.
            boef2 naar-x: boef2 x? y: speler y?.
            m wekker: 2 over: 1000.
      }.
      (nummer = 2) ja: {
            boef1 naar-x: boef1 x? y: speler y?.
            boef2 naar-x: speler x? y: boef2 y?.
            m wekker: 1 over: 1000.
      }.
}.
m wekker: 1 over: 0, scherm: ['achtergrond.png'].
```

Optioneel huiswerk/werkstuk

Leerlingen kunnen op eigen gelegenheid het spel verder uitwerken door bijvoorbeeld:

- muziek toe te voegen
- geluid bij botsingen toe te voegen
- de speler en boeven te animeren (meerdere afbeeldingen)
- het spel zo aan te passen dat er telkens een nieuwe level wordt gemaakt
- score bij te houden en tonen op het scherm
- een titelscherm te maken

(1 van deze als huiswerk, in groepjes of individueel)