# Module User Experience | Gaming

## Onderdeel Gameplay Activiteit 3: Mechanics, vier elementen

## Auteurs

Paul Bergervoet (Gameplay en teamleiding)

Monique Dewanchand (C#)

Lars Tijsma (Unity)

Anton Visser (Gameplay en testen)

# 

# 

# Activiteit 3: Mechanics ontwerpen met...

Er zijn allerlei zaken die je kunt inzetten om mechanics te ontwerpen. We gaan er hier vier bekijken die heel vaak gebruikt worden:

* **Objecten**, alle ‘dingen’ die je op de een of andere manier kunt gebruiken
* **Tijd**, snelheid...
* **Toestand (state)**, verschillende toestanden waarin iets kan zijn
* **Toeval**

Deze vier elementen komen al in kleine games vaak allemaal voor.

Eerder keken we naar Tetris, een simpel spel. De *objecten* in Tetris zijn de zes configuraties van vier vierkantjes. *Tijd* is belangrijk doordat de blokjes steeds sneller vallen. *Toeval* bepaalt welk blokje het volgende is. De *toestand* van blokjes verandert als ze landen, je kunt ze dan niet meer draaien of verplaatsen.

Aan het eind hebben we het ook nog heel kort over een vijfde element:

* **Surprise**,de verrassing.

## -- Objecten in het spel

Bij de acties die je als speler doet, gebruik je vaak objecten: gereedschappen, wapens, pick-ups die je punten of levens geven, schatten die je moet verzamelen, obstakels, enzovoort. De game zelf stuurt allerlei soorten objecten op je af: kogels en wapens die je verwonden, 43 soorten zombies, …

Soms gaat het om enkele simpele objecten: In Pacman heb je pilletjes (punt), sterren (veranderen Pacman in jager) en soms fruit (veel punten). Daarnaast heb je vier monsters. In totaal zijn er dus niet meer dan zeven verschillende soorten objecten en game characters.

Soms is er een parade van objecten met steeds net iets andere eigenschappen. Sommige spellen leven van deze variatie. Een goed voorbeeld is *Plants vs Zombies*. In dit spel ben je de trotse eigenaar van een huisje met tuin. Iedere nacht dringen de zombies je tuin binnen op weg naar je huis. Gelukkig kun je je tuin vol zetten met zombie-werende planten, ieder level komt er weer een nieuwe soort bij. Elk van die planten heeft een eigen manier om zombies te bestrijden en ook een andere prijs. Doel van het spel is om zo te planten, dat je de zombies buiten de deur houdt.



*Plants vs. Zombies*

Het kiezen van de planten die je in de tuin zet om de zombies buiten te houden, is het belangrijkste spelelement in Plants vs Zombies. Daarom is het belangrijk dat er echt iets te kiezen is. Daarvoor moet je de keuzes in *balans* brengen.

### Balans

Als voorbeeld kijken we naar de wapens in een fictief schietspel. Wat die precies zijn, doet er niet zo toe, het gaat erom dat je je wapens kunt kiezen. Er zijn drie belangrijke eigenschappen van de wapens: *kracht*, *kosten* en *beschikbaarheid van munitie*. Je ziet ze in de tabel hieronder.

|  | **Kracht** | **Kosten** | **Munitie voorraad** |
| --- | --- | --- | --- |
| **Wapen 1** | 100 | 10 | 100 |
| **Wapen 2** | 65 | 30 | 45 |
| **Wapen 3** | 35 | 60 | 55 |

Het is duidelijk: *Wapen 1* is het beste. Het is krachtig, kost weinig en er is genoeg munitie. Je kunt haast geen andere keuze maken!

Eigenlijk: er is helemaal geen keuze! Iedereen neemt natuurlijk *Wapen 1*, de rest kun je net zo goed uit het spel laten. Zo’n keuze die altijd de beste is, heet ook wel een *dominante strategie*. Een truc om altijd te winnen...

De keuze van wapens is uit balans. Om dit op te lossen, moeten we een uitspraak van Johan Cruyff toepassen: “*Elk nadeel heb z’n voordeel*.” De wapens moeten voors en tegens hebben, als speler moet je dan afwegen wat je het liefst hebt. Wanneer we de getalletjes veranderen:

|  | **Kracht** | **Kosten** | **Munitie voorraad** |
| --- | --- | --- | --- |
| **Wapen 1** | 100 | 80 | 40 |
| **Wapen 2** | 65 | 50 | 40 |
| **Wapen 3** | 45 | 50 | 100 |

Je ziet de spelers nu al denken: “*Wapen 1 is wel fijn, maar duur! De andere zijn goedkoper en Wapen 2 is dan nog best krachtig, maar Wapen 3 heeft weer het voordeel dat je zo makkelijk aan munitie komt.*”

Er is nu weer een echte keuze, het spel is beter in balans. Balans zit bij spellen in nog veel meer zaken, we komen ze nog tegen.



*Inventory in Pine*

#### Opdracht GD.7: Balans, voors en tegens van objecten

Bekijk nog eens je favoriete game X. Is er hier sprake van verschillende soorten objecten die voor hetzelfde doel gebruikt kunnen worden, zoals wapens of power ups?

Zo ja, beschrijf kort hun eigenschappen en geef aan hoe game X tot balans komt in de keuze van deze objecten.

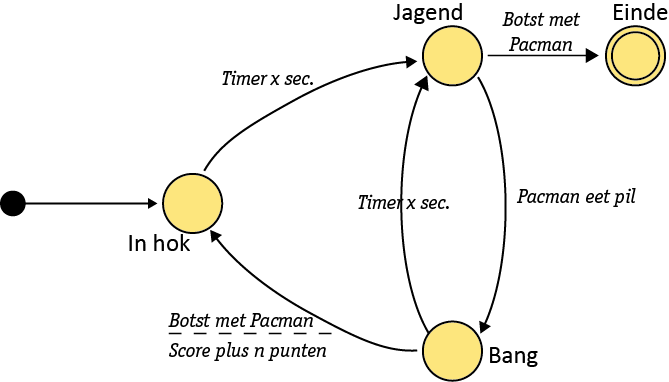
Zet het resultaat in de presentatie van de game-analyse van game X.

## -- Toestand (state)

Objecten, vijanden en ook de speler zelf reageren niet altijd hetzelfde. Een speler kan bijvoorbeeld tijdelijk onkwetsbaar worden, omdat hij/zij ergens een powerup heeft opgepakt. De regels veranderen daardoor. Een hit door een vijandelijke kogel geeft dan geen schade meer.  
We zeggen wel dat zo’n powerup de *toestand* van de speler verandert. Vaak wordt hier de Engelse term gebruikt: *state*.

Toestand zit in veel meer dingen: de hoeveelheid *health* die je nog hebt, bijvoorbeeld. Ook objecten kunnen toestand hebben: een bom staat op scherp of niet, een lamp is aan of uit, …

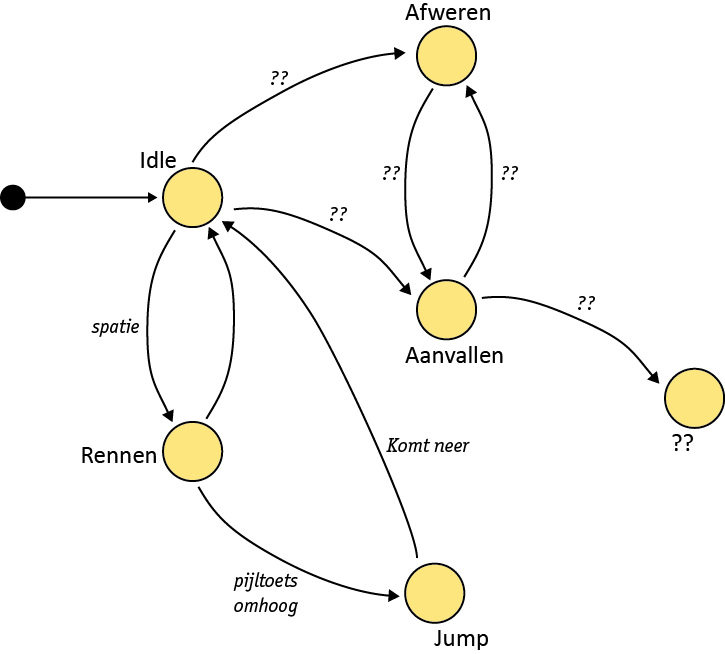
Net als het gebruik van een menigte verschillende objecten kan het werken met toestanden van objecten de variatie   
Eerder in het vak informatica heb je geleerd dat je toestandsdiagrammen kunt gebruiken om mogelijke toestanden en toestandsovergangen te beschrijven. Zo kun je precies beschrijven hoe een monster in Pacman werkt: een monster begint in het hok, na een tijdje gaat’ie jagen op Pacman, het level is ten einde als het monster de Pacman vangt. Als Pacman een speciale pil eet, wordt het monster bang, enzovoort.



*Toestanden van een monster bij pacman*

Zo’n toestandsdiagram kun je ook gebruiken om goed in kaart te brengen wat de mogelijkheden van de speler zijn. De speler kan stilstaan, lopen, rennen, springen, ontwijken, aanvallen. Dan kun je bedenken wat de mogelijkheden zijn. Kun je direct vanuit een ontwijk-actie aanvallen? Kun je niet springen als je stilstaat? Of juist niet als je rent? Wat wordt je toestand na een sprong?

Hieronder zie je een deel van het toestandsdiagram van een speler die met toetsen bestuurd wordt. Wanneer je dit aanvult, kun je precies duidelijk maken wat wel en wat niet kan.



*Toestandsdiagram van de speler*

#### Opdracht GD.8: Toestandsdiagram van een speler

Misschien heeft jouw game X ook een speler die in verschillende toestanden kan zijn, zoals hierboven. Zo nee, kies dan even een andere game (met zo’n speler) die je kent.

Maak een toestandsdiagram voor die speler en voeg dit toe aan de presentatie van de game-analyse.

*NB: het zou kunnen zijn dat je speler zeer veel toestanden heeft. In dat geval kun je je beperken tot een stuk of zes toestanden, naar keuze.*

## -- Tijd

In veel games speelt tijd een belangrijke rol. Tijd kan op allerlei manieren een rol spelen.

* Vijanden verschijnen in een bepaald tempo;
* Naarmate het spel vordert komen de vijanden steeds sneller;
* Een shield (onkwetsbaarheid) werkt maar een tijdje;
* Je beweegt met een snelheid, snelheid is niets anders als afstand per tijdseenheid;
* Je score is afhankelijk van hoe snel je een uitdaging oplost;
* Je moet een level binnen een bepaalde tijd uitspelen.

Tijd is zo belangrijk, omdat het voor een groot deel de speelervaring bepaalt. Is het tempo te hoog, dan wordt het spel onspeelbaar en haken de spelers af. Als het tempo te laag is, dan vallen de spelers in slaap en haken ze ook af. Het is heel lastig om uit te vinden wat goed is. Een vuistregel: begin rustig en voer de snelheid op als de speler het goed doet. Of dat zo is, kun je aflezen uit het aantal punten dat de speler scoort, bijvoorbeeld.

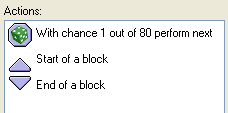
Wanneer het tempo goed is, kun je tijd gebruiken om de spanning op te voeren. Zo neemt de snelheid waarmee de blokjes in Tetris vallen langzaam toe. Het spel wordt zo steeds moeilijker en spannender.  
Het verhogen van de snelheid is een vorm van *progressie* in een spel. Dat is het onderwerp van activiteit 4.

  *Scene uit Pine*

## -- Toeval

Stel je speelt een spel twee keer vanaf het begin. De tweede keer merk je dat er eigenlijk precies hetzelfde gebeurt als de eerste keer. Dat is soms prima: als je een level in een platformgame niet haalt, heb je meestal wel een idee hoe het beter moet. Als de tweede keer dan alles door toeval anders is, schiet je daar niet veel mee op.  
Vaker is het leuker wanneer er niet precies hetzelfde gebeurt. Het toeval speelt een rol bij het verschijnen van nieuwe vijanden bijvoorbeeld. Toeval kun je dus gebruiken om variatie in een game mogelijk te maken.

Game engines hebben ingebouwd gereedschap om toeval te maken. Je gebruikt dan een zogenaamde **random generator**. Een **random** getal is een willekeurig getal, meestal onder een opgegeven waarde. Zo kun je vragen om een random getal *x* onder de 10 (of een andere waarde). Daardoor kun je bijvoorbeeld een random vijand laten spawnen: als er 10 soorten vijanden zijn, neem je vijand type *x*.

*Toeval in GameMaker*

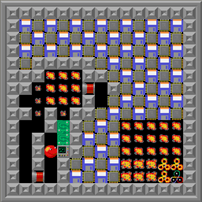
## -- Surprise

Tja, we gaan hier niet al te veel over zeggen. Als we hier de surprise-elementen uit jouw game-ontwerp gaan vertellen, is de verrassing er wel af…

De regels hierboven maken duidelijk dat het niet mogelijk is om een recept hiervoor te geven. Een recept voor verrassing?

Om er toch *iets* over te zeggen: Verrassing kan zitten in gameplay elementen, in ‘wat objecten doen’.

Een voorbeeld is het spel Supaplex dat uit 111 levels bestaat. Je komt in elk level een nieuwe verrassing tegen. De ene keer zijn het kanonskogels die je moet ontwijken, de volgende keer is er een schaartje dat je dood kan knippen, in een volgend level krijg je ineens schijfjes die je kunt gebruiken om een toegang ergens te forceren, omdat ze blijken te ontploffen als ze de diepte in vallen. Het werkt verslavend. In een [gameplay-filmpje](https://youtu.be/PqQj1QPT4ts) zie je het voorlaatste level en alle trucs die je moet uithalen om naar het einde van het level te komen.

*Supaplex*

Bedenk dat de verrassing uiteindelijk in het hoofd van de speler zit: er gebeurt op een zeker moment iets dat de speler niet verwacht. Je kunt proberen te spelen met de verwachtingen van spelers. Dat kan de spelervaring enorm verrijken.



*Doe iets onverwachts!*

#### Opdracht GD.9: Tijd, toeval en surprises in game X

In je favoriete game X zijn misschien ook spelelementen waarin tijd en/of toeval een rol spelen. Zo ja, geef van elk een voorbeeld. Voeg dit toe aan de presentatie.

#### Opdracht GD.10: Mechanics in je game-ontwerp

Het is moeilijk voor te stellen dat *geen* van de elementen hierboven in je eigen spelontwerp voorkomt.

Breid je spelontwerp uit met spelelementen waarin *objecten, state, tijd, toeval* en/of *surprise* een rol spelen. Mogelijk zitten deze al in je ontwerp en kun je deze checken op zaken als balans.

*Bedenk: het ontwerp kan groter zijn dan het demo level dat je gaat bouwen. Je hoeft niet alles in je ontwerp ook echt te bouwen! Ook is het niet nodig dat alle elementen voorkomen, dat hangt af van de aard van het game concept.*

Gebruik bij deze opdracht bijvoorbeeld de lenzen 4 (*Surprise*), 27 (*Time*), 28 (*State Machine*), 31 (*Action*) en 36 (*Chance*) uit de app van Jesse Schell.