# Module User Experience | Gaming

## Onderdeel Gameplay Activiteit 4: Progressie en flow

## Auteurs

Paul Bergervoet (Gameplay en teamleiding)

Monique Dewanchand (C#)

Lars Tijsma (Unity)

Anton Visser (Gameplay en testen)

# 

# 

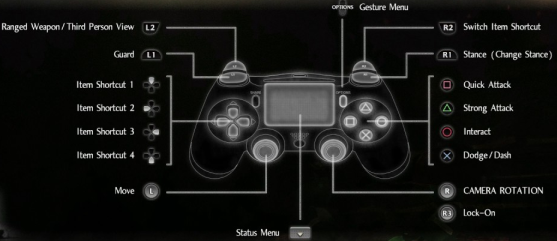
# Activiteit 4: Training, progressie en flow

Sommige spellen hebben een heel simpele mechanic. Als speler ontdek je vanzelf hoe het werkt. Dat is bijvoorbeeld bij Pacman. Je ziet gelijk dat je punten krijgt als je de pilletjes eet en je ontdekt snel genoeg dat de monsters gevaarlijk zijn :).

Bij spellen met uitgebreide mechanics moet de speler ‘leren’ hoe de gameplay werkt.

Daarvoor zijn er en paar mogelijkheden:

* **Uitleg**: er is een *instructiefilmpje* of een serie *slides* die uitleggen hoe het werkt. Ook een scherm met instellingen werkt zo. Wanneer je kunt instellen dat ‘pijl omhoog’ springen is, dan weet je daardoor ook *dat* je kunt springen.
* **Oefenen**: het spel heeft een *training level.* Je loopt als speler door het level en soms verschijnt er een pop-up die vertelt over een actie die je kunt doen. Je krijgt dan uitleg over de besturing. Op dezelfde plek is een simpele uitdaging die met die actie gedaan kan worden.
* **Puzzelen**: de speler moet zelf ontdekken hoe dingen werken. Goede spellen geven dan hints, op een subtiele manier.

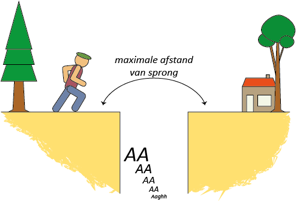


*Instellingen van een game: hoe bestuur je het met een controller?*

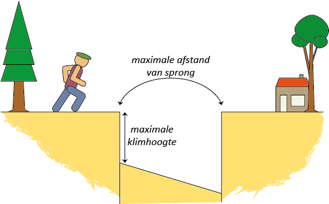
Op de handleidingen gaan we hier niet in. We gaan hier kijken naar hoe je een spel zo kunt maken dat een speler de gameplay ontdekt en kan oefenen.

## -- Train de speler

Er zijn nogal wat spellen waarin je moet springen. Of je nu eerst uitleg krijgt over het springen, of ontdekt dat het nodig is (je staat voor een kloof), het springen gaat niet vanzelf goed. Je kunt een (eerste) level zo opbouwen dat de speler de vaardigheid oefent. Dat kan bijvoorbeeld zo:

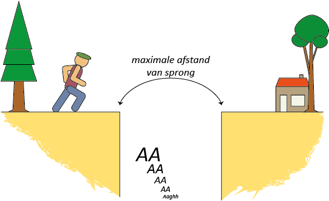


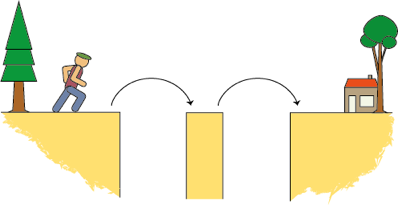
*Het eerste sprongetje is makkelijk,   
veel korter dan de maximale afstand*

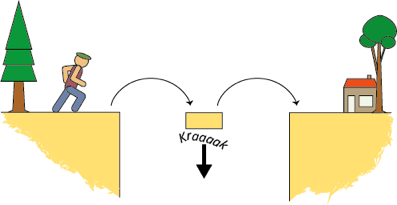


*Deze is lastiger, maar je hebt een uitweg.*

*Je kunt terug klimmen.*

*Nu voor het echie! Gevaar!*

*Dan naar de ‘double jump’...*

*Nu snel, het platform stort in!*

## -- Laat de speler ontdekken

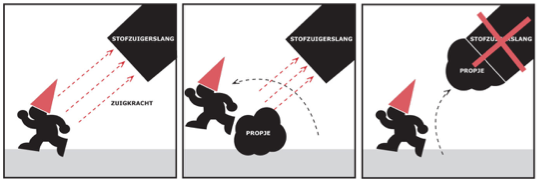
Onderdeel van de spelervaring is dat een speler keuzes heeft en zelf ontdekt dat een aanpak beter werkt dan een ander. Hierdoor word je als speler steeds beter en heb je de ervaring van het succes.

Dit kan bijvoorbeeld in een shooter. Je ontdekt dat het lopen van de ene route veel gevaarlijker is dan het lopen van een ander. Je ontdekt plekken waar je even dekking kunt zoeken. Zo kom je uiteindelijk veel beter een level door.

Soms kun je met de normale gameplay nieuwe elementen in een spel brengen. Als ontwerper van het spel kun je dan hints in het spel inbouwen, waardoor de speler dit spelelement makkelijker kan ontdekken.

Een voorbeeld komt uit het uit het ontwerp van het spel Guerilla Gnomes, gemaakt door studenten van de HAN (Hogeschool van Arnhem en Nijmegen). In dit spel bestuur je kabouters, die het Witte Huis willen overnemen, omdat ze het niet niet eens zijn met de president. Het Witte Huis is een gevaarlijke plek voor kabouters. Je kunt bijvoorbeeld zomaar opgezogen worden door een stofzuiger. ‘GAME OVER’….

Daar is wat tegen te doen. Als je een prop papier voor de stofzuiger gooit, dan raakt ie even verstopt en kun je ervandoor gaan.



Hoe werkt de hint hier? Dat is eigenlijk heel simpel. De eerste keer dat de stofzuiger verschijnt, is dat op een plek waar veel propjes zijn. Die stofzuiger kan dan vanzelf een keer een propje opzuigen en verstopt raken. Zo breng je de speler op een idee.

#### Opdracht GD.11: De lerende speler

We keren terug naar je favoriete game X, voor de game-analyse.

1. Is er in game X sprake van het trainen van de speler? Zo ja, beschrijf een voorbeeld.
2. Zijn er in game X ook spelelementen die je als speler moet ontdekken? Zo ja, geef daarvan een voorbeeld.
3. Verwerk dit in een slide in de presentatie over de game-analyse.

## -- Progressie: de opbouw in een spel

De zaken die eerder in dit stuk behandeld werden, gaan eigenlijk over de opbouw van een spel. Dat is een belangrijk punt, omdat je in een spel een goede opbouw moet hebben. Dat is nodig om de spelervaring op peil te houden.

De reden is simpel. Als je een spel speelt, dan oefen je dat. Je wordt er steeds beter in. Het spel moet daarin meegaan.

Bij spellen waarin je tegen de computer speelt, is dat lastiger dan bij fysieke spellen of sporten. Neem bijvoorbeeld tennis. Iemand die al vijf jaar intensief speelt, heeft een heel ander niveau dan iemand die pas vijf weken speelt. Je kunt dan eigenlijk niet goed tegen elkaar spelen.   
Je lost dat op door tegenstanders van gelijk niveau op te zoeken. Het is dan een heel ander spelletje dan tegen een beginner, maar belangrijk: aan de spelregels hoef je niets te veranderen. Die zijn voor beginners en pro’s hetzelfde.

Wanneer je tegen de computer speelt, oefen je ook. Je wordt steeds beter in het spel. Als de computer dan steeds hetzelfde spelletje speelt, op hetzelfde niveau, wordt het al snel vervelend.

Wat je wilt, is dat het spel langzaam maar zeker lastiger wordt.

In computerspellen wordt dat op allerlei manieren gedaan.

* **Snelheid verhogen**.  
  Dit is een veelgebruikte, simpele manier om een spel lastiger te maken. Het zit in veel spellen, juist ook in de simpele. Neem Tetris, dat spel wordt steeds lastiger doordat de blokjes steeds sneller vallen.
* **Meer, groter, sterker.**  
  Je moet verder springen, er komen steeds meer vijanden op je af, die zijn ook steeds sterker...
* **Meer objecten, wapens, hulpmiddelen.**  
  Je krijgt nieuwe wapens, planten, schatten, en ….
* **Nieuwe spelelementen.**Je voltooit een level en ontsluit nieuwe mogelijkheden in je character...

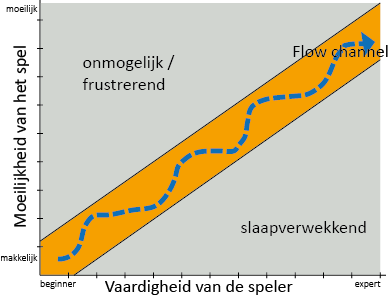
Veel manieren om zo’n opbouw te realiseren hebben we in de activiteiten over mechanics al besproken. We hadden het al over tijd (dus snelheid) en toeval (met grote en kleine kansen). Die kun je gebruiken om de uitdagingen groter te maken.

We hadden het over steeds grotere sets van objecten en de balans die je daarbij moet bereiken. Ook daarmee maak je een spel ingewikkelder en leuker voor de ervaren speler. We hadden het tenslotte over training en het introduceren van nieuwe spelelementen.

### Flow

Het einddoel van de ontwerper is: vind de juiste balans tussen *de speler die steeds beter wordt* en *uitdagingen die steeds lastiger worden*. Als dat lukt blijft de speler enthousiast.

Als dat lukt, kan de speler in een staat van *flow* terechtkomen. Het spel ontwikkelt zich zo dat je het precies kunt bijbenen, succes behaalt en totaal in het spel opgaat!



*Flow*

Dit plaatje laat de flow zien: wanneer de vaardigheden van de speler passen bij de moeilijkheid, dan zit je in de oranje strook, het *flow-kanaal*. Bij de grijze gebieden is dat niet zo.

Linksboven is te moeilijk: de speler haalt steeds maar een uitdaging niet en haakt gefrustreerd af. Rechtsonder is te makkelijk: het spel wordt vervelend, want je haalt een uitdaging met twee vingers is de neus.

De blauwe pijl laat zien hoe een spel zich ideaal ontwikkelt. De lijn slingert een beetje. Dat is opzettelijk. Het werkt beter als je een moeilijk stuk afwisselt met een wat makkelijker stuk, een *rustmoment*.

Even een concreet voorbeeld: Het spel ‘1945’ is een eenvoudige scrolling shooter die gemaakt is in GameMaker. Je bestuurt een vliegtuig, onderin het scherm. Van boven komen de vijandelijke vliegtuigen op je af. Schiet ze neer en ontwijk ze!

Omdat het spel in GameMaker gemaakt is, kun je gemakkelijk wat getalletjes veranderen en zo de snelheid van het spel aanpassen.

In het eerste plaatje zie je het begin, alles gaat nog langzaam…

*1945, gaap...*

Ok, dit is leuk voor tien seconden, als je even de besturing wilt oefenen. Moet niet te lang duren, je zit zo rechtsonder het flow-kanaal!

*1945, lijkt wel 1945000, help!*

Hmm, een beetje doorgeschoten naar de andere kant. Vermoedelijk kun je als speler oefenen wat je wilt, hier kun je nooit tegenop. Zit duidelijk linksboven het flow-kanaal.

Er is maar één regel om uit te vinden hoe je de moeilijkheidsgraad moet opvoeren: maak een schatting voor versie 1 van je spel en ga *playtesting* doen. Alleen wanneer je spelers bezig ziet, kun je echt vaststellen of je tempo goed is!

#### Opdracht GD.12: Progressie in Game X

We keren terug naar je favoriete game X, voor de game-analyse.

1. Is er in game X ook sprake van progressie, bijvoorbeeld door het opvoeren van de moeilijkheidsgraad. Geef daarvan een aantal voorbeelden.
2. Verwerk dit in een slide in de presentatie over de game-analyse.

#### Opdracht GD.13: Voortgang in je game-ontwerp

We keren terug naar het ontwerp van je eigen game. Deze opdracht kun je doen zoals het hieronder staat, maar je kunt ook de workshop op de volgende pagina doen.

1. Bekijk hoe je spelelementen in je eigen game introduceert. Bedenk of je aan je ontwerp elementen kunt toevoegen die de speler helpen.
2. Zit er vooruitgang in de game, gaan de uitdagingen van makkelijk naar moeilijk? Welke elementen voeg je toe om dit te realiseren?

Gebruik bij deze opdracht bijvoorbeeld de lenzen 21 (*Flow*), 34 (*Skill*) en 38 (*Challenge*) uit de app van Jesse Schell.



*“Hier eindigt de theorie over het ontwerp van de gameplay”*

# 

# Workshop Level Design

Veel van het voorgaande ging over de vraag hoe je uitdagingen verdeelt over een level. Laat je de speler oefenen? Zit er progressie in de uitdagingen? Zijn er keuzes en zijn die keuzes in balans?

Het is moeilijk, misschien zelfs onmogelijk, om in één keer een level te ontwerpen waarin dit allemaal goed is. Daarom wordt er vaak gedaan aan *paper* *prototyping*. Je maakt een level op een groot vel papier. Met poppetjes of plakkertjes zet je de uitdagingen neer. Daarna ga je in discussie over het ontwerp. “Is de progressie goed?”, vraag je elkaar en ook andere vragen van hetzelfde soort. Je schuift met de poppetjes als het nog niet goed is. Later ga je testen.

## Een prototype voor een Roll-a-Ball

In deze workshop ontwerp je een variant van Roll-a-Ball, met speciale aandacht voor de onderwerpen in deze activiteit.

We gebruiken hierbij een groot vel papier, de wereld. Met sticky’s geven we de obstakels en objecten in de wereld aan waarop de gameplay zich concentreert. Eventueel kun je een deel van de wereld tekenen met viltstift (nadeel is dan dat je dat niet zomaar verandert).

**Stap 1: de wereld en pickups**

Begin met het ontwerpen van een deel van of de hoofdlijn van de wereld. Zijn er ‘kamers’, of wanden waar je niet door kunt? Bedenk kenmerken van de spelwereld die de gameplay beïnvloeden.

Pickups geven de speler niet alleen punten, voor de ontwerper is het bron van het creëren van uitdagingen: plaats een pickup op een ‘lastige’ plek. Plaats de pickups in je level.

**Stap 2: speleracties: jump**

Een mogelijke uitbreiding op de speleracties is de jump.

Breidt het level uit, zodanig dat de speler met de jump dingen kan die anders moeilijk of onmogelijk zijn.

Denk ook over de introductie van deze actie.

**Stap 3: een extra mechanic -- lek en ….**

Hoewel een spel met een rollende bal simpel lijkt, zijn ook hier allerlei uitdagingen mogelijk die het spel spannender en leuker maken. Bijvoorbeeld: in het spel zijn plekken me gevaar, de bal kan bijvoorbeeld lek raken. Dat is vooral leuk als de speler daar ook iets tegen kan doen...

Bedenk een complete mechanic rond dit gegeven en bouw deze in het level in.

**Stap 4: een eigen mechanic**

Bedenk een geheel eigen toevoeging aan de gameplay.