# Module User Experience | Gaming

## Onderdeel Gameplay Activiteit 0: Introductie tot de module

## Auteurs

Paul Bergervoet (Gameplay en teamleiding)

Monique Dewanchand (C#)

Lars Tijsma (Unity)

Anton Visser (Gameplay en testen)

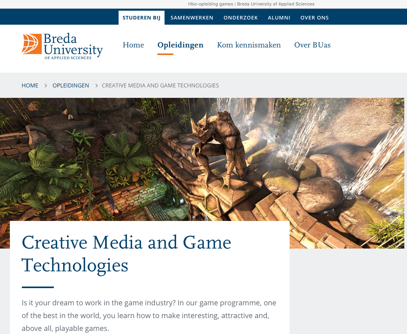
# 

# 

# Module Game Design

Welkom bij de module Game Design in het vak informatica. In deze module leer je over het ontwerpen en bouwen van games. Dat wil zeggen, je leert een deel van het vakgebied kennen.

Game Design is namelijk een groot vakgebied. Veel onderwerpen in de informatica zie je er terug, van digitaal artwork tot aan (stevig) programmeren en kunstmatige intelligentie. Game design omvat ook veel onderwerpen buiten de informatica, zoals kunst, psychologie en muziek. Het vakgebied is zo groot, dat er een aparte opleiding voor is aan de Breda University of Applied Sciences: Creative Media and Game Technologies.



*Site van de opleiding Game Technology, te vinden via* [*buas.nl*](https://www.buas.nl/opleidingen/creative-media-and-game-technologies)

## -- Aanpak

In deze inleiding maak je eerst kennis met Game Design in de volle breedte, we bekijken wat er voor een grote game allemaal nodig is. Veel daarvan zullen we in deze module niet kunnen doen. We gaan ons richten op twee onderdelen: het ontwerpen van spelelementen en het bouwen/programmeren van een spel in Unity (of eventueel GameMaker).



*Logo van Unity*

In deze module gaan we ervan uit dat je in een project werkt. Je ontwerpt in een team een spel en gaat dit bouwen (of gedeeltelijk bouwen). Het lesmateriaal in deze module dient om dat project te ondersteunen.

Bij het werken aan deze module ben je niet alleen bezig met het ontwerpen van je eigen game. Veel elementen willen we laten zien aan de hand van bestaande games. Je zult regelmatig opdrachten krijgen om een game te bekijken met de blik van een ontwerper: “*hoe zit het in elkaar?*” Kies voor die opdrachten een game uit die je goed kent en die je graag speelt. We raden aan om bij die opdrachten steeds dezelfde, favoriete game te gebruiken: *game* ***X***. Die opdrachten zien er dan meestal zo uit: "*Analyseer voor game X ...*"

#### Opdracht INTRO.1: Je favoriete game

1. Kies een game die je zelf speelt en die tot je favoriete games behoort: *game X*. Het mag een computerspel zijn, maar een bordspel of een ander spel zonder computer mag ook. De game hoeft niet per se een grote game te zijn, wat kleiner kan ook (al is boter-kaas-en-eieren wel erg klein).
2. Open de presentatie GameAnalyse\_GameX.ppt. Deze presentatie kun je gebruiken om de resultaten van je game-analyse vast te leggen - en later te presenteren. Zet als eerste de naam van je game X op de eerste slide.

## -- Spelervaring

Het kortste stukje uit deze module, maar misschien wel het belangrijkste onderwerp. Je denkt een spel te ontwerpen, maar uiteindelijk gaat het om de spelervaring van de speler. Een goede educatieve game kan een student een goede leerervaring geven, maar meestal gaat het om het amusement. Een spel is mooi, of spannend, of gewoon leuk om te spelen.

Dat is natuurlijk subjectief, iedereen heeft hierin een eigen voorkeur. Het duidelijkst is dat wel bij schietspellen. Sommige mensen worden er niet goed van, andere zijn niet van de spelcomputer los te rukken. Daardoor is het nauwelijks mogelijk hier iets algemeens over te zeggen. Wat je wel kunt is aangeven waarom jij (persoonlijk) een game leuk vindt.

*Belangrijk*: het gaat dan vooral om de *spelervaring*, niet om het uiterlijk of iets dergelijks. We zijn dus niet op zoek naar opmerkingen als “Dit doolhofspel is zo leuk omdat er prachtige schilderijen aan de muren hangen (en verder niks).”

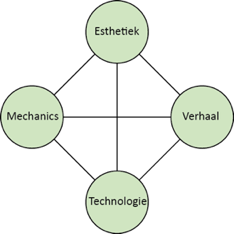
#### Opdracht INTRO.2: De spelervaring van je game X

1. Geef aan wat je favoriete game, je *game X* zo leuk maakt. Wees daarin zo specifiek mogelijk: geef bijvoorbeeld aan waarom game X beter is dan vergelijkbare spellen.
2. Vul de eerste slide in je ppt-presentatie in.

Hoe bereiken we die ervaring? We kijken naar vier aspecten.

## -- De vier pijlers van games

Jesse Schell noemt in zijn boek en app (zie kader) vier pijlers van het game design, vier zaken die altijd in games aanwezig zijn. Deze vier pijlers zijn **esthetiek** (*aesthetics*), **mechanics** (*idem*), **verhaal** (*story*) en **technologie** (*technology* :). Deze pijlers hebben allemaal met elkaar te maken.



*De 'elemental tetrad’ van Jesse Schell*

De vier pijlers zul je in elke game aantreffen, al is het soms maar een klein beetje. We bekijken ze apart.

| **A Book of Lenses**  *Jesse Schell is een bekende schrijver over het ontwerpen van games. Zijn boek “*The Art of Game Design, a Book of Lenses*" wordt in game-opleidingen veel gebruikt. In dit boek bekijkt hij het ontwerp van een game steeds door een andere bril (dit zijn de lenzen van de titel).*  *Bij zo'n lens horen steeds vragen, die je jezelf als ontwerper zou moeten stellen. Door de vragen te beantwoorden, kom je op dingen die je ontwerp beter maken.*  *Een voorbeeld is de lens van de speler. Daarbij vraag je je af voor welke spelers je je game maakt en welke spelelementen bij die spelers passen. Die zou je dan in je ontwerp moeten opnemen.*  *Een van de mooie dingen van het boek van Jesse Schell is dat de auteur zijn lenzen in een app gezet heeft. Deze is gratis te downloaden in de App Store (IPhone) of de Google Play Store (Android). Alle lenzen staan er in. Niet schrikken, het zijn er meer dan honderd. Je hoeft ze niet allemaal te gebruiken, natuurlijk. Blader eens en kies een paar vragen die je helpen.*  App van Jesse Schell |
| --- |

### (1) Esthetiek

Esthetiek gaat erover hoe een spel eruit ziet, hoort, ruikt, enzovoort. Bij een (fysiek) spel of sport gaat het om het bord, het speelveld, de vormgeving van de gebruikte attributen en ....

Bij computerspellen gaat het om het *beeld* en het *geluid*, inclusief *muziek*. Die moeten mooi zijn, passend bij de aard van het spel, of doelgericht. Esthetiek gaat nu eenmaal over het mooi zijn, de schoonheid van zaken. Dit terrein heeft zich in de afgelopen 50 jaar enorm ontwikkeld. Van heel simpele graphics bij de oudste (en minst krachtige) computers tot realistische werelden nu. Een paar belangrijke termen:

Spellen met simpele graphics gebruiken plaatjes van een paar honderd pixels om de game-objecten (speler, vijanden, power ups, pick ups, enz, enz) te laten zien. Zulke plaatjes noemen we **sprites**. Vaak zitten ze samen in een soort vellen, **sprite sheets**.



*Sprite sheet (stukje)*

De figuurtjes in een spel kun je laten bewegen door filmstripjes van sprites te maken. iedere keer dat het spel opnieuw op het scherm gezet wordt (gerenderd wordt), ga je naar het volgende plaatje. Daardoor gaat de pacman hieronder happen.

=

*Pacman-frames*

Bij de nieuwe, grote spellen gaat het er heel anders aan toe. Om te beginnen is het in 3D, dan heb je niet meer genoeg aan een plaatje voor een figuur. In 3D maak je een draadmodel - een **static mesh** - dat je van alle kanten kunt bekijken. Dan heb je verder nog een een speciaal plaatje nodig, meestal **texture** genoemd, dat je om de mesh heen kunt vouwen, zodat het er echt als een figuur of voorwerp uitziet.



*Game character en het draadmodel*

Beweging wordt dan ook een stuk ingewikkelder. Poppetjes hebben een soort van intern skelet, een **rig**, met botten die kunnen bewegen. Bij het weergeven van de mesh moet je dan wel precies bepalen hoe de mesh vervormt als bijvoorbeeld een arm beweegt.

Kortom, dit is behoorlijk ingewikkeld. En dan hebben we het nog niet over de speciale effecten, zoals *materials* (bijvoorbeeld bewegende oppervlakken als water en lava) en *particle systems* (rook, waterval, sneeuw).

Al die technieken kun je gebruiken om een eigen visuele sfeer te maken. Wordt het een *realistische* wereld, *horror*, *cartoony* of iets speciaals als *steampunk*?



*Beeld uit de game* Pine *van het Nederlandse Twirlbound*

**Game art**, characters en animaties zijn krankzinnig veel werk, als je ze echt mooi wilt maken. Dit kun je bijvoorbeeld zien in de credits van een grote game als Horizon Zero Dawn (zie kader ‘Rollen in een game project’, verderop). De lijst heeft grote 'departments' voor art, sound, music, visual effects en animation. Het maken van 'human animation' is een specialisme, het maken van een compleet personage met uiterlijk en bewegingen is vele honderden uren werk!

In deze module gaan we geen werk maken van het creëren van game art. In je game kun je gebruik maken van bestaande game art, muziek en geluiden. Daarvan is voldoende te vinden op internet. Zie bijvoorbeeld de [asset store](https://assetstore.unity.com/lists/top-free-3d-models-5063) van Unity. Een site met onnoemelijk veel gratis geluidjes is [Freesound](https://freesound.org).

Software om 3d-art te maken is vaak duur. Gratis software voor eenvoudige 3D-art is [SketchUp](https://www.sketchup.com). Met [Blender](https://www.blender.org) kun je complexe game-art maken.

| **Game Art en sfeer**  *Iedereen kent het spelletje Pacman. Het is een klassieker die nog een paar keer in deze module voor zal komen.*    *Pacman van het Oostenrijkse bedrijf* [*Masswerk*](https://www.masswerk.at/JavaPac/JS-PacMan2.html)  *In het kader van het eerstejaarsproject bedachten studenten van de Universiteit Utrecht een bijzondere variant van het spel. Ze maakten er een 3D-spel van, met een first-person perspectief. Dat wil zeggen dat je als speler door de ogen van Pacman kijkt, je bent als het ware zelf Pacman. De doolhof werd een nachtelijke stad, met een beetje griezelige verlichting. De monsters gaven een beetje licht, zodat je ze om de hoek kon zien aankomen.*  *Maar: verder waren alle spelregels precies hetzelfde: op straat lagen pilletjes die je kon eten, er lag een ster om de monsters tot prooi te maken, enzovoort.*  *Deze verandering veranderde de spelervaring totaal. Het werd een zeer griezelig en spannend spel, terwijl alleen het uiterlijk veranderde. Dit laat zien dat de esthetiek van een spel zeer belangrijk is. Het is (mede) bepalend voor de spelervaring, met name het gevoel dat het spel oproept.* |
| --- |

### (2) Verhaal (story)

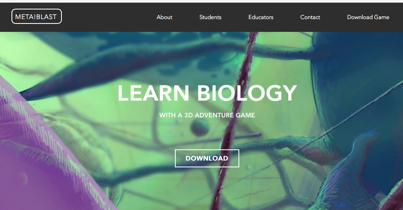
Het verhaal van een game is een veel kleiner element, maar het is wel belangrijk. Zelfs de kleinste games hebben iets van een verhaal. Neem Pacman, dat is niet een deel van een cirkel van wisselende grootte, dat tegen puntjes botst. Nee, het is een happertje, een persoontje, dat pilletjes eet en op de vlucht is voor monsters. Zo'n verhaal bij een spel helpt ons om ons in te leven in de hoofdpersoon!

Bijna alle spellen hebben een verhaal, ook bijvoorbeeld educatieve spellen. Het spel *Metablast*

is een adventure game waarin je leert hoe cellen in planten werken. De game begint met: *“It is 2051. Our earth's ecosystem is reverberating as some unknown pathogen decimates plant life. A team of expert scientists has inexplicably disappeared while investigating the cause. Can you rescue the lost team, discover what is killing the plants, and save the world?”*

*(Het is 2051. De wereld schudt op zijn grondvesten omdat een onbekende ziekte alle planten vernietigt. Een team van wetenschappers is vermist, terwijl ze op onderzoek waren. Kun je het team redden, de ziekteverwekker opsporen en de wereld redden?)*

Zo’n verhaal trekt je direct de game in!



*De educatieve game Metablast*

Daarnaast heb je natuurlijk games die draaien om een verhaal. Adventure games en Role playing games hebben een groot verhaal dat je stap voor stap moet ontdekken.

Een belangrijke voortrekker op dit gebied is het spel *Myst*, uit de jaren tachtig. Wanneer het spel start, land je op een eiland zonder een enkele aanwijzing wat er gebeurd is. Langzaam maar zeker ontrafel je het verhaal.

Myst is ook een stap in game art. Computers waren nog niet krachtig genoeg om in een 3D-wereld rond te lopen. Daarom was de wereld eigenlijk een slideshow, maar wel een heel mooie!



*Myst*

Ook de pijler 'verhaal' is een vak apart, het valt buiten de scope van deze cursus. Vind je dit leuk, bekijk dan zeker het maken van een heldenverhaal: de ‘[*Hero’s Journey*](https://en.wikipedia.org/wiki/Hero%27s_journey)’. Dit laat verschillende versies zien van de opbouw van een heldenverhaal, van de ‘call’, de roep(ing), tot de ultieme test.

En: gebruik je fantasie om een verhaal te bedenken bij je eigen game.

| **Het verhaal van Space Invaders**  *Ook heel kleine games hebben vaak een verhaal. Zo is er een bekend stukje geschiedenis rond het spel Space Invaders. Space invaders is een van de oudste computerspellen, uit het eind van de jaren 70 van de vorige eeuw. De speler, onderin het scherm, schiet op de ruimteschepen van aliens, die langzaam naar het aardoppervlak afdalen.*    *space invaders*  *In die tijd kwamen de eerste 'video games' op die in speelhallen gespeeld konden worden (niemand had thuis een computer). Daarbij zaten ook schietspellen en die lokten toen al discussie uit over geweld in spellen. Bij space invaders was het eerst de bedoeling dat je ook zou schieten op legers van soldaten. Om de discussie over geweld te vermijden werden de soldaten vervangen door aliens. Deze kleine verandering in het verhaal hielp om het spel een succes te maken.* |
| --- |

#### Opdracht INTRO.3: Esthetiek en verhaal van game X

Hoewel we er niet dieper ingaan, is het wel goed om game X kort op deze punten te bekijken.

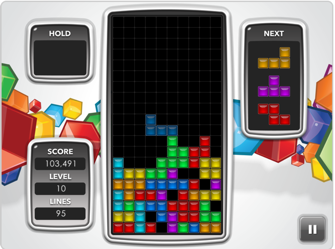
1. Esthetiek. Wat maakt jouw favoriete game X mooi. Kun je iets zeggen over de stijl van het artwork in het algemeen. Zijn er bijzondere dingen aan het uiterlijk van de speler, andere game characters, de wereld, enz. Hoe zit het met geluid en muziek.
2. Verhaal. Geef in drie of vier zinnen een korte samenvatting van het verhaal achter game X.
3. Vul de powerpoint-presentatie over game X aan.

### (3) Mechanics

Fijn, we komen nu bij de pijlers die we wel gaan uitwerken.

**Mechanics**is een belangrijk woord in de wereld van game-ontwerpers. Bij de mechanics gaat het om het belangrijkste van de game: hoe werkt het spel? Laten we eens een simpel spel bekijken.

Bij het spel *Tetris* heb je blokjes die opgebouwd zijn uit vier vierkantjes. Die blokjes vallen naar beneden in het verder lege speelveld. Als speler kun je de blokjes naar links en naar rechts schuiven en je kunt ze draaien, met de pijltoetsen. Op de bodem van het speelveld moet je de blokjes zo in elkaar passen dat er volle regels ontstaan. Die verdwijnen dan en daarvoor krijg je punten. Doel is om zo veel mogelijk punten te halen voor je af gaat, omdat het speelveld helemaal vol met blokjes is. De blokjes vallen steeds sneller, dus er komt een moment dat je het niet meer bijhoudt...



*Tetris, te spelen via tetris.com/play-tetris*

We kunnen nog wel dingen preciezer maken, maar eigenlijk is het met deze zes regels tekst wel duidelijk. Ze beschrijven heel behoorlijk ‘hoe het spel Tetris werkt’. De mechanics van Tetris zijn dus het schuiven en draaien van blokjes door de speler, het vallen van blokken die neerkomen op blokjes die al onderin liggen en het weghalen van volle regels.

Nu is Tetris een simpel spel. Grotere spellen hebben meer mechanics, maar in het algemeen is er toch één die het meest gebruikt wordt. Denk bijvoorbeeld aan:

* Platform game: je beweegt horizontaal en springt om op platforms te komen. Daarbij moet je obstakels en vijanden vermijden en andere objecten juist oppakken.
* Candy Crush / Bejeweled: maak rijtjes van gelijke stenen of snoepjes.
* First Person Shooter: Je speelt een soldaat of krijger die door een gevaarlijke wereld

loopt waar de hele tijd tegenstanders opduiken. Schiet de tegenstanders neer!

* Enz. enz. enz...

In deze module zullen we veel dieper ingaan op mechanics. We beginnen bij het doel van een spel: wat moet je als speler bereiken om het spel te winnen of uit te spelen? Verder gaat mechanics over uitdagingen in het spel. Wat kun je doen? Wat zijn de regels die dan de uitkomst bepalen? Hoe zorg je dat het spel niet saai of onmogelijk wordt?

| **Een nieuwe mechanic met schieten**  *Het is moeilijk om een compleet nieuwe mechanic te bedenken. Wereldwijd gebeurt dat maar één of enkele keren per jaar. Maar een compleet nieuwe bedenken is ook helemaal niet nodig. Je kunt al heel wat bereiken door een leuke variant op een mechanic te bedenken. Zo bedachten studenten van de HAN (Hogeschool van Arnhem en Nijmegen) een mooie variant op een schietspel.*  *In hun spel Erios speel je Cupido/Eros, het hulpje van Venus/Aphrodite, de godin van de liefde uit de Grieks-Romeinse mythologie. Je weet misschien dat Cupido beschikt over een kleine pijl en boog, waarmee hij liefdespijlen kan afschieten. Dat klinkt behoorlijk waardeloos, wanneer er in de game een paar monsters op je afkomen.*  *Maar wacht even... Je schiet eerst een pijl op het eerste monster, daarna op het tweede. Je hebt vervolgens geen centje last meer van ze, ze hebben alleen nog maar oog voor elkaar! De liefde is te sterk...*  *Op dat idee kun je voortbouwen. Even later komt Cupido een open brug tegen, hij kan er niet over. De knop om de brug neer te laten is aan de overkant, daar kun je niet bij... Maar: schiet een liefdespijl op een monster aan de overkant, daarna een op de knop en voila!*  *Hé, hé, eindelijk een schietspel dat niet bloederig is…*  Concept art van het spel Erios, studenten van de HAN |
| --- |

#### Opdracht INTRO.4: Mechanics van game X

1. Wat is het voornaamste spelelement - mechanic - in jouw game X? Vergelijk zonodig met het lijstje van mechanics dat hierboven staat.
2. Heeft je spel ook afwijkende spelelementen? Zo ja, noem er enkele.
3. Vul de presentatie over game X aan.

### (4) Technologie

Bij technologie denk je natuurlijk direct aan computers, maar er is veel meer. Gewone spellen gebruiken ook technologie. Denk bijvoorbeeld aan een tennisracket. Vijftig jaar geleden was een tennisracket nog een houten ding met een klein blad, bespannen met snaren uit darm. Tegenwoordig is een racket high tech en worden rackets steeds verbeterd zodat je er harder en preciezer mee kunt slaan.

Maar goed, dit is het vak informatica en we kijken dus vooral naar computertechnologie. Ook dan zijn er allerlei technische apparaten, bijvoorbeeld *gamecontrollers* en *virtual-realitybrillen*. Daar is in deze module geen ruimte voor.



*Gamecontroller*

De technologie waar we hier mee gaan werken, heet een **game engine**. Een game engine is een stuk standaard software dat allerlei dingen voor je regelt, zodat je dat als gamebouwer zelf niet mee hoeft te programmeren. Een game engine doet dingen die in bijna alle games voorkomen, steeds op min of meer dezelfde manier. Een voorbeeld is *collision detection*, het detecteren van botsingen.

Dat kan op allerlei manieren in games zitten. Het kan zeer onaangenaam zijn (je wordt geraakt door een vijandelijke kogel en valt dood neer) of juist zeer aangenaam (je raakt/pakt een gouden pil en wordt onkwetsbaar). Het kan een actie zijn (je raakt/drukt op een knop en een deur gaat open). Voor de game engine maakt het allemaal niet uit: die controleert gewoon of twee objecten elkaar raken, of ze ongeveer op dezelfde plek zijn.

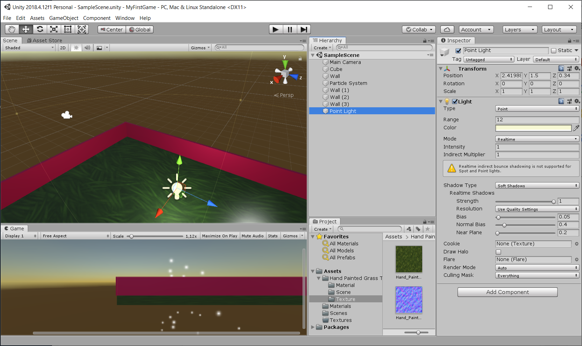
Collision detection is best lastig om te programmeren. Omdat het in de game engine zit, hoef je dat niet zelf te doen. Als game-programmeur hoef je dan alleen nog maar te programmeren wat er bij een botsing moet gebeuren.

Er zijn meer dingen die een game engine levert:

* **Physics**. Zorgt ervoor dat dingen gaan zoals in de echt wereld. Er is zwaartekracht, dingen vallen. Een bal stuitert, maar een een schep hutspot niet. Je glijdt van een hellend vlak af…
* **Rendering**. De game engine zorgt ervoor dat het spel op het scherm getoond wordt. De wereld wordt vanuit de huidige positie weergegeven, met achtergrond, game objecten en meer. Bij 2D-games is dit niet zo ingewikkeld: teken alle sprites op de juiste plek. Bij 3D is het lastiger. Iets dat ver weg is moet klein getekend worden, dichtbij is groot. Er zijn misschien lampen, schaduwen en meer. Belichting is heel moeilijk voor een game engine, het kost heel veel rekentijd.
* **Tools**. Game engines hebben heel veel tools om het maken van een gamewereld makkelijker te maken. Je kunt dan bijvoorbeeld makkelijk een speelveld met muurtjes, achtergrond en zo maken. Vaak zijn er allerlei speciale mogelijkheden, bijvoorbeeld *speed trees* (bomen die bewegen in de wind) en *particle systems* (rook, watervallen, vulkanen…)

Deze zaken worden allemaal toegepast in de **game loop**, zie kader.

In deze module richten we ons op de game engine **Unity**, omdat dit een uitgebreide engine is waar je mooie 3D-games kunt maken.



*Unity*

| **De Game Loop**  *De kern van een game engine is de game loop. Wanneer een spel loopt, is de game engine bezig met het herhalen van de stappen in een game. Na elke stap wordt de game opnieuw getekend,* gerenderd *op het scherm. Bij elke stap wordt er gekeken wat er in de game veranderd is.*  *Bijvoorbeeld: In een schietspel gebeurt elk iedere stap van alles. De speler verplaatst zich, vijanden en kogels ook. Een kogel kan iets raken, er valt iets. Er wordt geschoten, dus er komen nieuwe kogels bij. Andere kogels verdwijnen juist, ze vliegen de wereld uit zonder iets te raken. Verder is er een brandje…*  *Al die veranderingen in de toestand van het spel worden aangeduid met de term* ***update.*** *Het spel en alle objecten hebben een methode* **update** *waarin staat wat er elke stap gedaan moet worden. Het tekenen (renderen) van het spel wordt gedaan door een methode* **draw***. De game loop van Unity ziet er dan zo uit:*    *In andere game engines loopt het iets anders. GameMaker heeft bijvoorbeeld een* ***Step Event****, waar je kunt aangeven wat er elke stap moet gebeuren. Wanneer je in GameMaker een game maakt, hoef je daar alleen de uitzonderlijke dingen op te geven. Botsingen, verplaatsingen, input en andere veel voorkomende zaken worden automatisch gedaan.* |
| --- |

#### Opdracht INTRO.5: De technologie achter game X

1. Heb je bij game X bijzondere attributen nodig (echt spel) of gebruik je bijzondere game-controllers (computergame)? Geef aan welke dit zijn.
2. Voor computerspellen: de meeste games worden gemaakt met behulp van een game engine zoals Unity. Probeer te achterhalen met welke game engine game X gemaakt is.
3. Vul de presentatie over game X aan.

Games kunnen enorm groot zijn. Je krijgt dan dat mensen allerlei gespecialiseerde klussen gaan doen en daar hun beroep van maken. Zo zijn er mensen die alleen maar bezig zijn met het maken van bewegende poppetjes. Deze mensen staan dan als *human animator* is de aftiteling, de credits van een game.

Daarnaast zijn er dan nog veel meer specifieke rollen, zie kader.

| **Rollen in een gameproject**  *Guerrilla Games (*[*www.guerrilla-games.com*](http://www.guerrilla-games.com)*) is een bekende, grote Nederlandse game-ontwikkelaar, vooral bekend van de game Killzone (en alle afgeleiden). In 2017 brachte Guerilla Games een compleet nieuwe game uit, waarvoor alles door het bedrijf zelf ontwikkeld was: Horizon Zero Dawn. Je kent het spel misschien.*  *Cover van Horizon Zero Dawn*  *Om een beeld te krijgen van wat er allemaal nodig is om zo'n game te bouwen, kun je de credits van Horizon Zero Dawn op imdb bekijken:* [*https://www.imdb.com/title/tt4044024/fullcredits*](https://www.imdb.com/title/tt4044024/fullcredits)*. De lijst begint met de acteurs, die vooral zijn ingehuurd vanwege hun stemmen, maar noemt ook alle ontwikkelaars, schrijvers, artists, programmeurs, 'human animators' en veel meer. Met uitzondering van de 'special thanks' ( :) ) zijn dit allemaal beroepen in de game-industrie.*  *Extra informatie over de ontwikkeling van HZD vind je op de Engelstalige Wikipedia:*  [*https://en.wikipedia.org/wiki/Horizon\_Zero\_Dawn*](https://en.wikipedia.org/wiki/Horizon_Zero_Dawn) |
| --- |

Naast de grote studio’s bestaan er ook veel kleine.

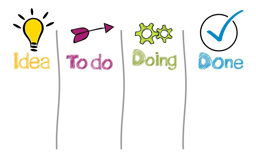
| **Een kleinere game studio: Twirlbound**  *Bij de Nederlandse game studio Twirlbound in Breda worden niet zo'n grote games gemaakt als bij Guerilla in Amsterdam. De studio van 7 mensen maakt hun games in Unity en gebruikt, zoals veel indie teams (zoals dat vaak genoemd wordt), meer externe hulpmiddelen en assets die ze niet zelf maken. Maar toch werkt Twirlbound aan games die wat groter zijn dan gemiddeld voor zo'n team: het genre waar ze zich op focussen is open world action adventure. Dat betekent dat het een grotere, aaneengeschakelde 3D wereld is waar de speler aan de hand van een avontuur doorheen gaat. Daarbij horen veel game mechanics, zoals exploration (rondrennen), action combat (vechten, in dit geval met een zwaard en schild) maar ook praten en onderhandelen met andere karakters in de gamewereld.*  *De Bredase studio maakt al zo'n 7 jaar games. Hun meest recente grote game heet Pine en kwam uit in oktober 2019 voor Steam en Nintendo Switch.*  *Pine*  *Op dat moment hadden de 7 mensen van Twirlbound al bijna 4 jaar aan Pine gewerkt. Het uitbrengen op verschillende apparaten is daarbij één van de grootste uitdagingen, omdat het op verschillende PCs verschillend werkt, en daarnaast ook voor een console moest werken (Nintendo Switch). Het is tegenwoordig dan ook gebruikelijk dat studios na het uitgeven van hun game nog fouten oplossen en dingen oppoetsen - dat heet patchen (vernoemd naar het plakken van een pleister).*  Door: Mathijs van de Laar, creative director van Twirlbound |
| --- |

## -- Het Gameproject

Stel je voor: je werkt bij een gamestudio, die redelijk succesvol is. Beter nog, je klas *is* een gamestudio. Jullie hebben pas een populaire game gepubliceerd, die goed geld opbrengt. Je hebt daardoor even de tijd om over een opvolger na te denken. De directie besluit dat de hele studio gaat meedenken over de volgende game.

De gamestudio (klas) wordt verdeeld in teams. Die teams gaan elk een idee - een game concept - opschrijven en uitwerken tot een compleet ontwerpdocument, een **game design document**. Van de game wordt een stuk(je) gebouwd. Zoals we al zagen is het bouwen van de complete game een enorme klus, daarom doen we nu maar een stuk, een **demo level**.

Aan het eind van het project worden de ontwerpen en de demo levels gepresenteerd (een pitch) en wordt er één gekozen.



De precieze opzet van dit project hangt nogal af van de omstandigheden op school, zoals omvang en tijd die voor het project is ingeruimd. Ook: havo/vwo, wel of geen Unity, ….

Je docent zal hier informatie over geven

## -- Verder in deze module

Plaatsen van de verschillende bronnen in de module:

* Introductie practica (video’s van Unity: Roll-a-ball)
* Game design activiteiten (uitleg en workshop)
  + Concept en doel
  + Mechanics en regels
  + Vier elementen bij mechanics
  + Progressie en flow: speler laten kennismaken met mechanic, training, progressie
* Practica over het maken van scripts in Unity. Dit zijn er ook vier, ze zijn gekoppeld aan game design activiteiten
  + Doel: verzamel pickups
  + Mechanics 1: jump
  + Mechanics 2: gevaren
  + Progressie: meer levels
* Aanvullend materiaal: C# voor Unity