

Dynamic Difficulty Adjustment

Houd de gamer in de Sweet Spot

Joost van Dongen

2 juni 2005

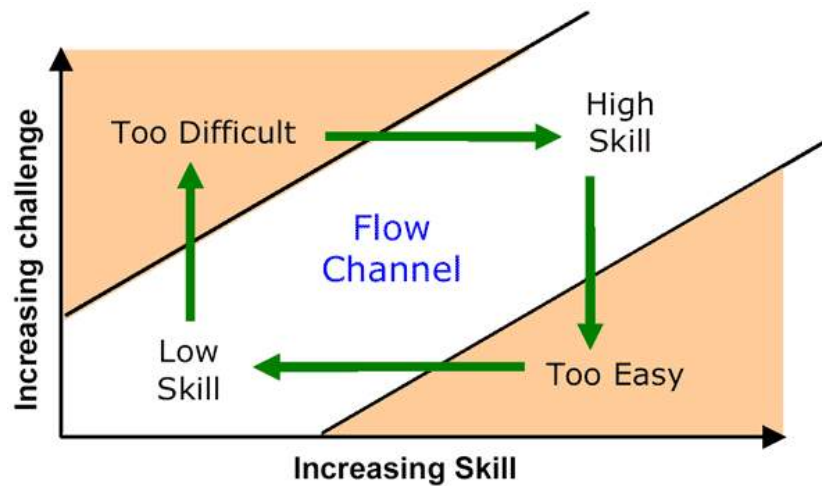
Website: www.oogst3d.net

E-mail: tsgoo@hotmail.com

1 De Sweet Spot

Ooit een game gespeeld die veel te moeilijk was? Waar je met geen mogelijkheid doorheen kwam? Die game heb je vast niet uitgespeeld en vond je vast niet leuk. Aan de andere kant: ooit een game gespeeld die veel te gemakkelijk was? Waar je zo doorheen liep? Je hebt die game misschien wel uitgespeeld, maar was dat echt leuk? Had je naderhand het gevoel echt iets bereikt te hebben met het uitspelen van die game? Vast niet. Ergens tussen deze twee uitersten in ligt de Sweet Spot: waar een game uitdagend is, maar niet onmogelijk, waar een level niet zonder moeite lukt, maar je ook telkens opnieuw moet beginnen. Hier is de game op zijn leukst om te spelen.

De game designer probeert altijd een game zo leuk mogelijk te maken, dus n van zijn taken is om zijn game precies in de Sweet Spot te krijgen. Dit kan gedaan worden door de game met dit in gedachte te bouwen en daarna te gaan testen: vinden gamers het leuk? Spelen ze het gemakkelijk door? Maar testen is duur en ook al test je, de ene gamer is beter dan de andere gamer en hier moet de game op aangepast worden. Daarom kun je bij de meeste games de moeilijkheidsgraad kiezen: easy, medium of hard voor de beginner, de gevorderde gamer of de fanaat. Maar hoe moeilijk is easy? In de ene game is easy misschien net zo gemakkelijk als medium in de andere game. Dus hoe moet de speler inschatten welk niveau die heeft? En sowieso: waarom moet de gamer zijn eigen niveau inschatten? Waarom kan de game dat niet doen, zodat de speler altijd in de Sweet Spot zit, ook als hij zichzelf te laag of te hoog inschat?



Figuur 1: *M. Csikszentmihalyi ontwikkelde dit flow model [1]. In de flow channel bevindt zich de Sweet Spot. Zit je onder de flow channel, dan is de game te gemakkelijk. Zit je erboven, dan is de game te moeilijk. Voor een ervaren speler bevindt de Sweet Spot zich op een hogere moeilijkheidsgraad dan voor een beginner.*

2 Dynamic Difficulty Adjustment

Een oplossing voor dit probleem is Dynamic Difficulty Adjustment, of DDA. DDA houdt in dat de game tijdens het spelen analyseert hoe de speler het doet. Komt de speler gemakkelijk door de levels heen, dan wordt de game wat moeilijker gemaakt. Gaat de speler vaak dood of is zijn health lang laag, dan wordt de game wat gemakkelijker gemaakt. Op deze manier wordt de moeilijkheidsgraad altijd aangepast zodra de gamer niet meer in de sweet spot zit.

Op de oppervlakte klinkt DDA geweldig. Kijk je echter wat dieper, dan komen er allerlei problemen naar boven. Hoe kun je het niveau aanpassen zo dat de speler het leuk blijft vinden? Hoe test je hoe goed de speler het doet? En is het eigenlijk überhaupt wel wenselijk om DDA in games te stoppen? Gaan gamers zich er niet enorm aan storen? In de rest van dit artikel zal ik proberen antwoorden te geven op deze vragen.

3 De moeilijkheidsgraad aanpassen

Er zijn vele manieren om de moeilijkheidsgraad aan te passen en de speler te helpen en vele daarvan zijn ronduit slecht. Zo is het mogelijk om bij een speler die lang weinig health heeft gewoon de health te verhogen. Echter, dit is dodelijk voor de spelervaring. De game probeert een overtuigende

wereld aan de speler voor te schotelen (niet pers een realistische wereld) en dan is het onzin dat er opeens spontaan een lading health bijkomt. Dit zal de speler uit de ervaring trekken, omdat het enorm opvalt dat hij wordt geholpen. Een ander voorbeeld waar het gemakkelijk mis kan gaan is bij het aanpassen van de kwaliteiten van een karakter. Dit kan als gevolg hebben dat in een basketbalspel bij een goede speler Jordan constant scoort als de computer hem gebruikt en telkens mist wanneer de speler hem gebruikt. De speler is tenslotte goed, dus de computer moet meer tegenstand bieden. Dit zal de speler het gevoel geven dat de computer valsspeelt, want waarom scoort die Jordan bij de computer wel en bij de speler niet.

De conclusie van deze twee voorbeelden is dat het aanpassen van de moeilijkheidsgraad met grote voorzichtigheid gedaan moet worden. Eigenlijk moet het zo gebeuren dat de speler het niet direct merkt. Hier zijn vele manieren voor. Ik zal hier een aantal methoden kort bespreken.

De eerste is om de wapens van de speler sterker en die van de tegenstander zwakker te maken, of juist andersom. Dit is gemakkelijk te doen, maar gaat snel opvallen: waarom was die slechterik de vorige keer met n schot te doden en vergt het nu drie schoten? Dit is dus geen goede methode. Beter wordt het om aan te passen wat voor wapens de tegenstanders hebben, of zelfs welke tegenstanders de speler tegenkomt. Wanneer de gamer in een nieuw gebied komt, weet hij niet van tevoren hoe zijn vijanden bewapend zullen zijn, dus zal de speler niet kunnen merken dat ze sterker of zwakker zijn gemaakt door andere vijanden neer te zetten of ze een pistool te geven in plaats van een machinegeweer. Een variant hierop is om afhankelijk van hoe goed de speler speelt een sidekick erbij te laten komen die de speler helpt, of om tegenstanders soms te laten slapen. Dit is niet alleen een simpele overwinning, maar ook een leuke variatie.

Een andere methode die goed werkt en die de speler amper doorheeft wordt geopperd door Scott Miller [2], maker van Max Payne. Hij zegt er het volgende over (waarbij ADD of Auto-dynamic difficulty hetzelfde is als DDA): “The idea of giving or withholding health items is another way of implementing ADD without the player being aware. We could have fairly easily done this in Max Payne by giving more or fewer bottles of pain killer, depending on the player’s situation when he came across a pain killer spawning point. Also, some thugs in Max drop pain killers randomly, and this could have easily been tied into the ADD system.”

Echt complex wordt het wanneer ervoor gekozen wordt om de tegenstanders slimmer of dommer te maken om de moeilijkheid aan te passen. Dit vergt een AI-systeem waarbij de slimheid van de vijanden instelbaar is. Dit is een heel subtiele en mooie oplossing, maar heeft een groot nadeel: zo’n goed AI-systeem is zelden voorhanden en wanneer je die speciaal hiervoor moet ontwikkelen is dat erg duur. Een bekend voorbeeld van een spel waarin dit gedaan wordt is Unreal Tournament, waar de bots zo ingesteld kunnen worden dat ze hun niveau aan de speler aanpassen.

En vrij exotische oplossing, maar ook een heel mooie, is om daadwerkelijk het level aan te passen aan het niveau. Als de speler erg goed is, moet hij langs een moeilijke route. Is hij een beginner, dan mag hij een gemakkelijker route spelen. Ook dit is erg duur, want dit betekent dat levels een stuk groter en ingewikkelder worden om te maken.

4 Meten hoe de speler het doet

Hierboven is besproken hoe de moeilijkheidsgraad aangepast kan worden, maar om dit te doen, moet je wel weten wanneer dit moet gebeuren en in welke mate. Hiervoor zal er een meetsysteem in de game ingebouwd moeten worden dat kan beoordelen of de gamer het goed of slecht doet. Dit lijkt in eerste instantie erg eenvoudig, maar hier duikt een lastig probleem op: veel spelers vertroebelen de meetresultaten door heel vaak te save en te laden, of leren hoe het meetsysteem werkt en gaan er dan misbruik van maken.

Scott Miller geeft hier een aantal voor de hand liggende dingen die gemeenten kunnen worden: de gemiddelde hoeveelheid energie die de speler heeft, het aantal slachtoffers dat de speler maakt (goede spelers laten doorgaans niemand over) en het aantal keer dat je sterft per level. Hij raadt tenslotte aan om niet het aantal afgeschoten kogels te tellen, want dit kan misleidend zijn. Denk bijvoorbeeld aan een goede speler die kogels genoeg heeft en daarom voor de lol in het rond gaat schieten.

Eigenlijk zijn al deze argumenten misleidend. Goede spelers willen misschien snel door het spel heen en laten daarom tegenstanders staan. Aantal sterftes per level en gemiddelde hoeveelheid health geven ook geen goed beeld, want veel spelers gebruik quick save en quick load zoveel, dat ze vrijwel nooit doodgaan of zwaar geraakt worden. Zodra dat gebeurt laden ze naar vlak daarvoor en spelen verder. De gemakkelijkste oplossing hiervoor is om met checkpoints te werken. Hierdoor kan de speler het systeem niet meer in de war brengen.

Hunicke en Chapman stellen in hun DDA-systeem Hamlet [1] een als variant een behoorlijk ingewikkeld systeem voor dat kijkt naar wat de speler allemaal heeft. Er wordt gekeken naar hoe de hoeveelheid health, kogels en power-ups die de speler in zijn bezig heeft variëren over de tijd en hier worden conclusies uit getrokken. Dit lost niet heel het quick save/load probleem op, maar wel gedeeltelijk. Een speler zal namelijk zelden herladen omdat zijn kogels bijna op zijn en dat wordt hierin wel meegerekend.

Hoe je het systeem ook maakt, het is bijna altijd mogelijk om het voor de gek te houden. Zo kan de speler er bewust voor kiezen om constant slecht te spelen, om zo een gemakkelijker eindbaas tegen te krijgen. Hier is niet echt een oplossing voor, want de speler kan altijd doen alsof hij slecht is. Daar staat tegenover dat het eigenlijk helemaal niet zo'n probleem is als de speler dit doet: hij mag met de game doen wat hij wil en als hij het zo leuk

vindt, dan is dat mooi. Het doel van de game is uiteindelijk toch om leuk te zijn, op welke manier dat ook is voor de gamer.

5 Is DDA wel wenselijk?

Nu duidelijk is hoe DDA toegepast kan worden, rest ons de vraag of we eigenlijk wel DDA willen toepassen. Het argument voor hebben we al gezien: een game is op zijn leukst als de moeilijkheid precies goed is en met DDA is dat bereikbaar.

Er zijn echter ook een aantal zeer sterke tegenargumenten. Cebrian zegt daarover het volgende in een reactie op Scott Miller's artikel [2]: "ADD introduces an element of variability at the rules that the player can easily identify as cheating. Players usually want a 'fair game', one with a fixed set of rules they can feel and understand: a bullet to the head will kill, three bullets will surely kill this particular enemy, and so on." Om deze reden is het nodig om heel erg voorzichtig te zijn met het toepassen van DDA. Hierdoor mag DDA eigenlijk niet toegepast worden op de regels van de game-wereld, omdat de speler verwacht dat die constant zijn.

Een ander tegenargument is dat die moeilijkheidskeuze aan het begin van het spel juist een reden is voor extra speelplezier. Een belangrijk aspect van gamen is de uitdaging die games bieden. Wanneer een gamer een spel heeft uitgespeeld, zal hij daarover vaak gaan opscheppen tegen zijn vrienden. Als de gamer weet dat het spel voor iedereen uitspeelbaar is, omdat het zichzelf gewoon zo gemakkelijk maakt als mogelijk, is het uitspelen veel minder leuk. Ook zijn die niveau's een reden voor herspeelbaarheid: als de speler het spel op medium heeft uitgespeeld, kan hij het daarna nog een keer op hard uitspellen. Met DDA vervalt deze mogelijkheid. Echter, hier zijn goede oplossingen voor. Zo kan ervoor gekozen worden om in het spel in een menu DDA uitzetbaar te maken, waarna de speler alsnog een vast niveau kan kiezen. De beginner merkt hier dan niks van, terwijl de fanatieke gamer deze optie wel zal vinden. Ook de oplossing van Max Payne werkt goed: normaal wordt het spel met DDA gespeeld, maar speel je het uit, dan komt er een moeilijker niveau beschikbaar waar geen DDA inzit.

Al met al zullen spelers het over het algemeen niet leuk vinden als het spel zichzelf gemakkelijker maakt als de speler niet goed genoeg is. Daarom is het handig om geheim te houden dat er DDA in de game zit. Spelers die het niet weten, kunnen zich er ook niet aan storen. Als DDA goed is uitgevoerd, zal ook slechts een klein percentage van alle spelers merken dat het erin zit.

De designer kan er ook voor kiezen om DDA juist heel zichtbaar te maken. De speler krijgt dan constant te zien op welk niveau hij aan het spelen is en zo kan het een uitdaging worden om het niveau zo hoog mogelijk te krijgen. Dit zal de herspeelbaarheid enorm verhogen.

Doordat er bij het inbouwen van DDA zoveel lastige keuzes komen kijken die allemaal een goed antwoord behoeven voor het maken van een goede game, is het met DDA nog steeds nodig om games uitvoerig te testen op moeilijkheidsgraad en zichtbaarheid van DDA. In eerste instantie leek het hoopvol dat dit niet nodig zou zijn, maar na alle valkuilen die tot nog toe zijn genoemd blijkt DDA zeker zo duur om te testen en uit te voeren als gewoon moeilijkheidsgraden.

Een laatste probleem met DDA dat nog genoemd dient te worden, is de verschillende speelstijl van verschillende spelers in verschillende buien. Veel spelers zullen inderdaad een uitdaging willen hebben in het spel, maar er zijn ook andere speelstijlen. Sommigen willen misschien gewoon even lekker knallen en zien hoe het verhaal loopt, maar willen het ondertussen niet moeilijk hebben. Anderen vinden het misschien juist erg leuk als het spel zo moeilijk is dat ze er amper doorheen komen: het overwinninggevoel als het toch lukt is dan veel groter. DDA kan niet detecteren waar de speler zin in heeft, dus dit schakel je eigenlijk voor een groot deel uit als DDA aanstaat. Daarom lijkt het toch wijselijk om DDA een optie te maken en niet alleen maar DDA in te bouwen.

6 Conclusie

Al met al is DDA een interessante techniek om een spel constant uitdagend te houden voor de speler. Er komen veel problemen bij kijken, maar die blijken allemaal goed oplosbaar. Helaas is n van deze oplossingen dat DDA uitgezet moet kunnen worden in het menu van de game, waardoor de designer alsnog zowel DDA als verschillende moeilijkheidsgraden moet inbouwen. Dit kost veel tijd en geld. Toch is DDA zeker een bruikbare techniek die in de toekomst vast in steeds meer games zal zitten.

Referenties

- [1] Vernell Chapman Robin Hunicke. Ai for dynamic difficulty adjustment in games.
www.cs.northwestern.edu/~hunicke/pubs/Hamlet.pdf
, 2004. (last visited on July 12, 2005).
- [2] Scott Miller. Auto-dynamic difficulty (article plus discussion).
dukenukem.typepad.com/game_matters/2004/01/autoadjusting_g.html
, 2004. (last visited on July 12, 2005).