

Stats / Damage Scaling & Balance

Studio Game Extended | Digital Ideation | Hochschule Luzern

Autor: Joel Christen
Betreut von: Dragica Kahlina, Sebastian Hollstein

Inhaltsverzeichnis

Einleitung	Seite 1
Untersuchung verschiedener Stats / Damage Scaling Modelle	Seite 2
Guild Wars 2	Seite 2
Rappelz	Seite 4
Champions of Norrath	Seite 5
League of Legends	Seite 7
Starbound	Seite 9
StarCraft II	Seite 11
Pokémon	Seite 13
Erkenntnisse	Seite 14
Quellen	Seite 14

Einleitung

Diese Arbeit behandelt die Frage, wie verschiedene Stats / Damage Scaling Modelle sich auf die Balance eines Spielverlaufs auswirken und welche Vor- und Nachteile die einzelnen Modelle aufweisen.

Es wird eine bestimmte Auswahl an Games aus verschiedenen Genres verglichen und analysiert und dabei versucht, deren Funktionalität zu bewerten, ohne die Kerndesignentscheidungen des Spieleentwicklers zu untergraben. Dabei werden jeweils die Formeln zur Schadensberechnung aufgezeigt, um Differenzen zu erkennen und je nach Game relevante Aspekte erläutert.

Es soll daraus eine Erkenntnis gewonnen werden, warum welche Modelle gut oder eben nicht gut funktionieren und ob das Genre beziehungsweise der Spielverlauf eines Games ein Modell bevorzugt und wenn ja, aus welchem Grund dem so ist.

Untersuchung verschiedener Stats / Damage Scaling Modelle

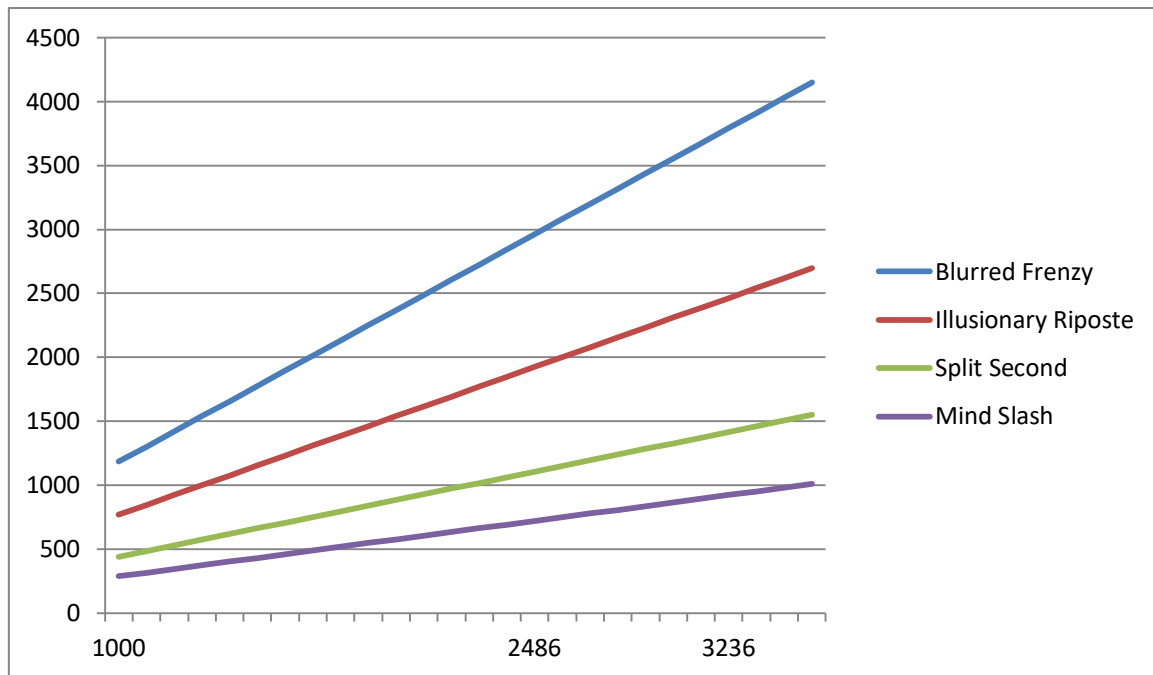
Game 1: Guild Wars 2 Genre: MMORPG (massively multiplayer online role-playing game)

Bei Guild Wars 2 gibt es zwei verschiedene Schadenstypen: Power Damage und Condition Damage. Power Damage ist direkter Schaden, der durch einen Armor Stat reduziert werden kann. Condition Damage sind DOTs (damage over time), welche den Armor Stat ignorieren. Diese Arbeit fokussiert sich auf das Scaling des Power Damages, dessen Schadensberechnungsformel wie folgt lautet:

$$\text{Damage} = (\text{Weapon strength} * \text{Power} * \text{Skill coefficient}) / \text{Armor}$$

Da es sich in dieser Formel ausschliesslich um Multiplikatoren handelt und Skills nur mit Koeffizienten und ohne Base Damage arbeiten, sind diese untereinander immer balanced. Skill A ist also prozentual gesehen stets gleich zu Skill B. Erhöht sich nun der Power Stat von Beispielsweise 2486 auf 3236 (also um 30%), so erhöht sich der Power Damage eines jeden Skills ebenfalls um 30%. Hierzu ein paar Skill Damage Vergleiche mit zwei unterschiedlichen Power Stats auf demselben Charakter:





Der Armor Stat skaliert ebenfalls prozentual, und stört die Balance zwischen den Skills daher ebenfalls nicht. Bei einem MMORPG wie Guild Wars 2, ist es von Vorteil, ein rein prozentuales Schadensberechnungsmodell zu benutzen, denn das Hauptspiel findet im End Game (Level 80 Charakter) statt und man kehrt nicht mehr ins Early Game (Level 1 Charakter) zurück, was eine Schiftung der Stärke der Skills untereinander im Spielverlauf obsolet macht.

Ein weiterer Vorteil dieses Systems ist die Möglichkeit des einfachen Skalierens der Stats. In Guild Wars 2 gibt es eine Level Spanne der Gegner in den einzelnen Gebieten. Befindet sich ein High Level Player sich nun in einem Low Level Gebiet, so wird der Charakter runterskaliert, um zu verhindern, dass der Spieler dort übermächtig ist. Dies erlaubt Guild Wars 2, alle Gebiete stets für Events und dergleichen zu nutzen, ohne dass sie für End Game Spieler uninteressant werden. Die Balance der Skills untereinander wird durch das prozentuale Scaling mit Power stets gewährleistet.



Game 2: Rappelz

Genre: MMORPG (massively multiplayer online role-playing game)

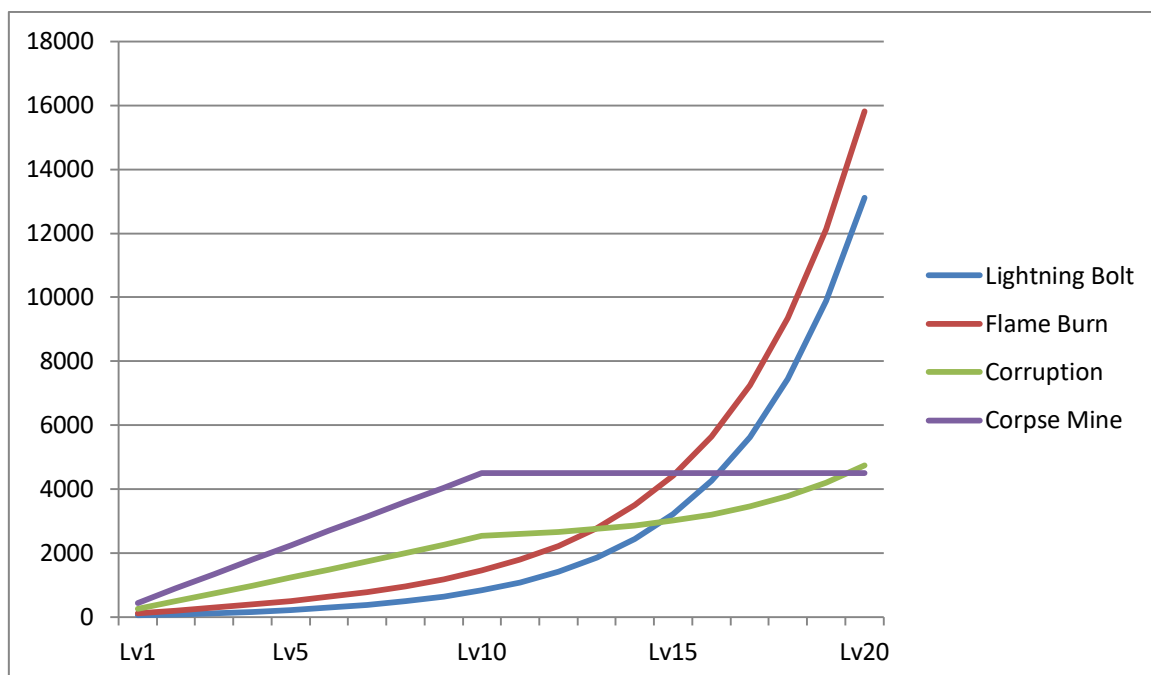
Bei Rappelz gibt es ebenfalls zwei Schadenstypen, Magic Damage und Physical Damage. Diese werden wie bei Guild Wars 2 durch einen defensiven Wert geschwächt. Obwohl die komplette Schadensberechnungsformel der Playerbase nicht bekannt ist, kennt man trotzdem die Teilformel der einzelnen Skills:

Magic Damage = (M.Atk * Skill coefficient) + Base damage

Physical Damage = (P.Atk * Skill coefficient) + Base damage

Hier wird nicht mit einem rein prozentualen System gearbeitet, denn alle Skills verfügen über einen Base Damage, doch einzelne Skills über kein multiplikatives Scaling mit M.Atk, beziehungsweise P.Atk und sind somit komplett Abhängig von ihrem Skill Level. Es werden hier vier Fähigkeiten des gleichen Charakters verglichen:

Lightning Bolt:	Damage = (M.Atk * (0.90 + 0.05 * Skill Lv)) + (15 * Skill Lv)	Max Skill Lv: 20
Flame Burn:	Damage = (M.Atk * (1.15 + 0.05 * Skill Lv)) + (70 * Skill Lv)	Max Skill Lv: 20
Corruption:	Damage = (M.Atk * 0.35) + (238 * Skill Lv)	Max Skill Lv: 10
Corpse Mine:	Damage = (450 * Skill Lv)	Max Skill Lv: 10



In diesem Diagramm ist ein exponentielles Wachstum an M.Atk miteinbezogen, um das Leveln in Rappelz zu simulieren. Durch das Diagramm werden einige Probleme des per Skill individuellen Schadensberechnungssystems sichtbar.

Alle Fähigkeiten, welche über kein M.Atk Scaling verfügen, gelangen nach dem Erreichen des Maximallevels an einen Endpunkt ihrer Stärke, daher werden diese früher oder später von anderen Fähigkeiten outscaled. Fähigkeiten, wessen M.Atk Modifikatoren nicht mit ihrem Skill Level skalieren, werden ebenfalls früher oder später von anderen Fähigkeiten outscaled. Fähigkeiten, deren Stärke zu Beginn prozentual gesehen aus einem grossen Teil aus Base Damage bestehen, werden je nach Nutzungseffizienz (z.B. Cast Time) unter Umständen outscaled.

Game 3: Champions of Norrath

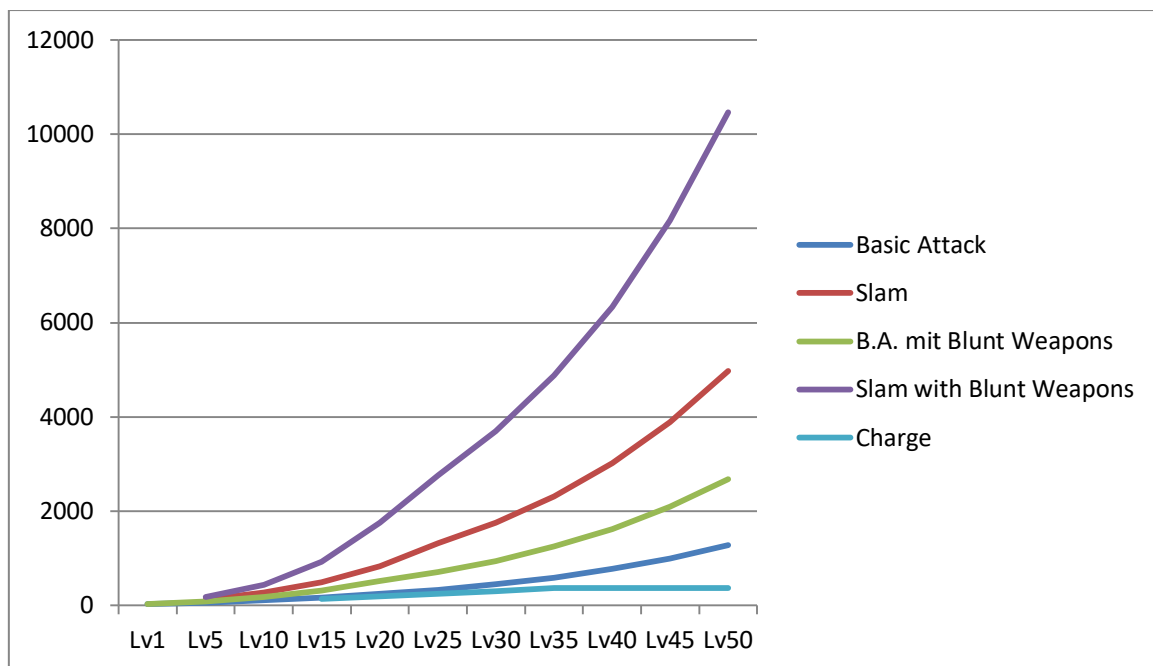
Genre: RPG (role-playing game)

Dieses Spiel verfügt über zwei Scaling Modelle, welche bei jedem Charakter je nach Fähigkeit angewendet werden. Einige Fähigkeiten sind prozentual Abhängig von der Basic Attack, andere bestehen aus einem festen Wert. Dies bringt massive Probleme mit sich, welche sich anhand eines Vergleichs zwischen den Skills des Barbarian Warrior's und des Erudite Wizard's darstellen lassen. Die leicht reduzierten Schadensberechnungsformeln lauten hier wie folgt:

$$\begin{aligned} \text{Basic attack damage} &= \text{Weapon strenght} * (\text{Strength} / 4) \\ \text{Skill type A damage} &= \text{Basic attack damage} * (\text{Modifier} * \text{Skill Lv}) \\ \text{Skill type B damage} &= \text{Base damage} * \text{Skill Lv} \end{aligned}$$

Der Barbarian Warrior verursacht den Hauptschaden mit Basic Attacks. Diese skalieren bei Champions of Norrath mit dem Strength Stat und mit der Weapon Strength, was ihn durch Leveln und neue Waffen automatisch im Spielverlauf bis zum maximalen Level (50) mitskalieren lässt. Er verfügt über passive Fähigkeiten wie „Blunt Weapons“, womit Schaden mit stumpfen Waffen prozentual erhöht wird. Dieser prozentuale Wert steigert sich ausserdem mit dem Level der Fähigkeit selbst, doch selbst wenn dem nicht so wäre, würde die Fähigkeit trotzdem mit nur einem Level relevant bleiben, da es sich um einen prozentualen Modifikator handelt. Ein anderer Skill nennt sich „Slam“ und ist vom oben genannten Typ A, also prozentual abhängig von der Basic Attack. Auch „Slam“ skaliert seinen prozentualen Modifier mit dem Level des Skills. In diesem Beispiel ist lediglich die Fähigkeit „Charge“ vom Typ B, also reiner Base Damage. Allerdings ist hier der Hauptgrund des Skills Utility, also Gegner zu unterbrechen oder ähnliches.

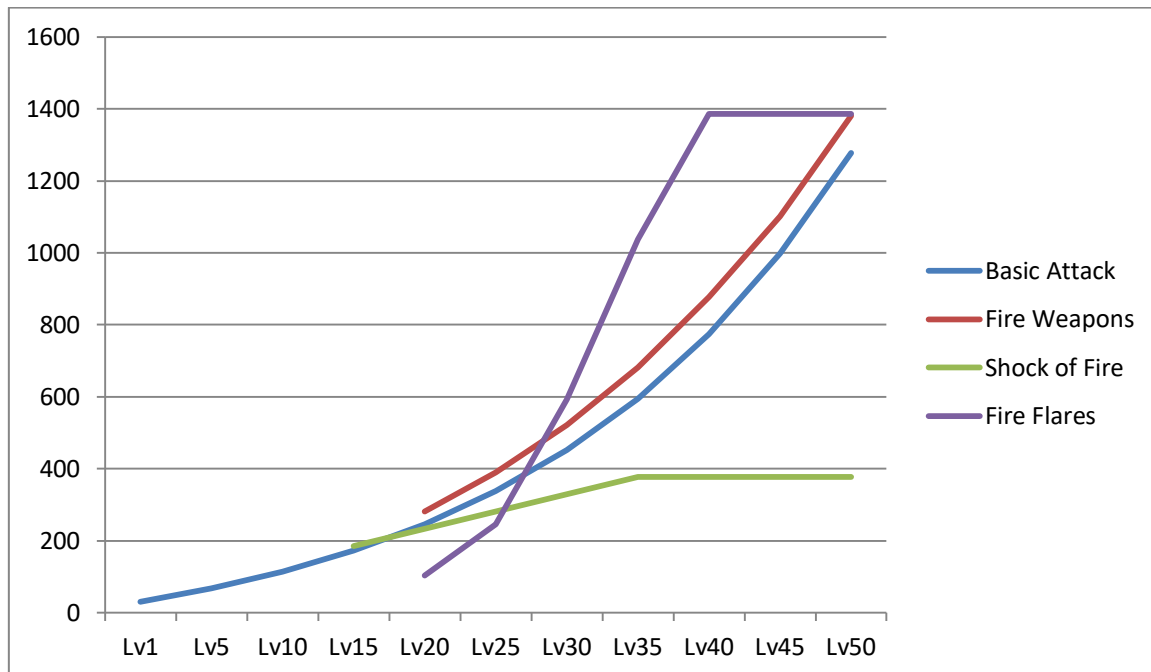
Basic Attack:	$\text{Weapon strength} * (\text{Strength} / 4)$	
Slam:	$\text{Damage} = \text{Basic attack damage} * (190\% + 10\% * \text{Skill Lv})$	Lv: 1-20
Blunt Weapons:	$\text{Damage} = \text{Basic attack damage} * (110\% + 5\% * \text{Skill Lv})$	Lv: 1-20
Charge:	$\text{Damage} = 144 + (12 * \text{Skill Lv})$	Lv: 15-35



Das Scaling des Barbarian Warriors funktioniert also von Level 1 bis 50 fast einwandfrei.

Der Erudite Wizard hingegen versucht seinen Schaden mit Fähigkeiten zu verursachen. Das Wort „versucht“ wird hier verwendet, da er ausschliesslich Skills vom Typ B besitzt und dieses Scaling Modell im Spiel so eindeutig nicht funktioniert. Er verfügt zwar ebenfalls über die Basic Attack, diese kann er aber nicht so effektiv verstärken wie der Barbarian Warrior. Sein Design wäre es, hauptsächlich mit Skills Schaden zu verursachen. Hier die Formeln der analysierten Skills:

Basic Attack:	$\text{Weapon strength} * (\text{Strength} / 4)$	
Fire Weapons:	$\text{Damage} = \text{Basic attack damage} + 18 + (17 * \text{Skill Lv})$	Lv: 20-40
Shock of Fire:	$\text{Damage} = 137 + (48 * \text{Skill Lv})$	Lv: 15-35
Fire Flares:	$\text{Damage} = (98 + (5 * \text{Skill Lv})) +1 \text{ additional hit per } 3 \text{ Skill Lv}$	Lv: 20-40



Hier kommen viele unterschiedliche Scaling Modelle zum Vorschein. Ein nur per Skill Level skalierender fester Stat als Basic Attack Verstärker, oder als separater Skill und ein fast nicht skalierender fester Stat, welcher via eines zusätzlichen Hits alle 3 Skill Level ein zweites Scaling erhält. Zudem stoppt deren Scaling sofort nach dem Maximieren des jeweiligen Skill Levels, was letztendlich dazu führt, dass selbst der Erudite Wizard mit Basic Attacks mehr Schaden verursachen wird, als mit seinen Skills. Ebenfalls problematisch ist der Fakt, dass der Stat „Strength“ den Schaden von Basic Attacks erhöht, der Stat „Intelligence“ allerdings nicht den von Skills, sondern das Mana und deren Regeneration erhöht, was den Spieler erlaubt, öfters Fähigkeiten zu benutzen, welche ohnehin zu schwach sein werden. Zudem wird es ebenfalls zu einem Punkt kommen, an dem die Regeneration des Manas, die Kosten des selbst stärksten Skills übersteigt und der Stat obsolet wird.

Strength increases melee damage and carrying capacity.

Intelligence increases total mana and mana regeneration.

Es kommen zwar noch Resistenzen und Schwächen gegen Elemente wie Feuer, Eis, Gift etc. in Spiel, welche der Erudite Wizard ausnützen kann, retten tun diese sein Scaling und die Balance allerdings nicht.

Game 4: League of Legends

Genre: MOBA (multiplayer online battle arena)

Doch sind feste Werte, also Base Damage nicht immer ein Unheil für das Scaling eines Games. Base Damage ist besonders dann gefragt, wenn ein Game in Partien gespielt wird und einige Fähigkeiten oder Charaktere im Early Game eine andere Stärke als im Late Game aufweisen sollen, dies also ein Designchoice ist. Ein gutes Beispiel hierfür ist das Game League of Legends. Die Charaktere besitzen Skills, welche meist sowohl über einen Base Damage wie auch einen Modifikator verfügen. Der angreifende Charakter (in diesem Spiel „Champion“ genannt) verfügt über offensive Stats wie Attack Damage (AD), Attack Speed, Critical Ratio, Critical Damage und Armor Penetration welche die Basic Attacks verstärken und Ability Power (AP), Cooldown Reduction und Magic respektive Armor Penetration, welche Skills verstärken. Das angegriffene Ziel verfügt über Health Points, Armor und Magic Resistance, welche dem entgegenwirken. Diese Stats können hauptsächlich nach und nach durch das Kaufen und Aufwerten von Items erhöht werden, doch skalieren einige auch mit dem Level (1-18) eines Champions. Um dieses funktionierende Schadensberechnungsmodell mit Base Damage zu erläutern, fokussiert sich diese Arbeit bei diesem Beispiel nur auf einen isolierten Champion und somit nur auf einen Teil der offensiven Werte, da es sich um ein sehr komplexes System handelt.

Es werden die Fähigkeiten des Champions „Viktor“ und deren Scalings miteinander verglichen:
(Bei Viktor handelt es sich um einen im Late Game starken Champion mit hohem Scaling)

Siphon Power:	Damage = $45 + (15 * \text{Skill Lv}) + (\text{AP} * 0.4)$	Lv: 1-5
Discharge:	Damage = $-5 + (25 * \text{Skill Lv}) + (\text{AP} * 0.55) + (\text{AD} * 1)$	Lv: 1-5
Death Ray:	Damage = $30 + (40 * \text{Skill Lv}) + (\text{AP} * 0.6)$	Lv: 1-5
Aftershock:	Damage = $-20 + (40 * \text{Skill Lv}) + (\text{AP} * 0.7)$	Lv: 1-5
Chaos Storm:	Damage = $25 + (75 * \text{Skill Lv}) + (\text{AP} * 0.5)$	Lv: 1-3
Chaos Storm Tick:	Damage = $25 + (40 * \text{Skill Lv}) + (\text{AP} * 0.45)$	Lv: 1-3

Ability Power
Ability power increases the amount of damage dealt with abilities.
Current Ability Power: **288 (0 + 288)**
You do up to **288** more damage with your abilities.

Siphon Power (5)
65 Mana
DPS: 482 [Q]
Cooldown: 4s

Viktor blasts an enemy unit, dealing 120 (+115) magic damage while granting Viktor a shield that absorbs up to 253 damage over the next 2.5 seconds.
Viktor's next basic attack deals 120(+158) bonus magic damage.
Augment - Turbocharge: Siphon Power's shield is increased by 60% and Viktor gains 30% Movement Speed for 2.5 seconds.

Ability Power
Ability power increases the amount of damage dealt with abilities.
Current Ability Power: **571 (0 + 571)**
You do up to **571** more damage with your abilities.

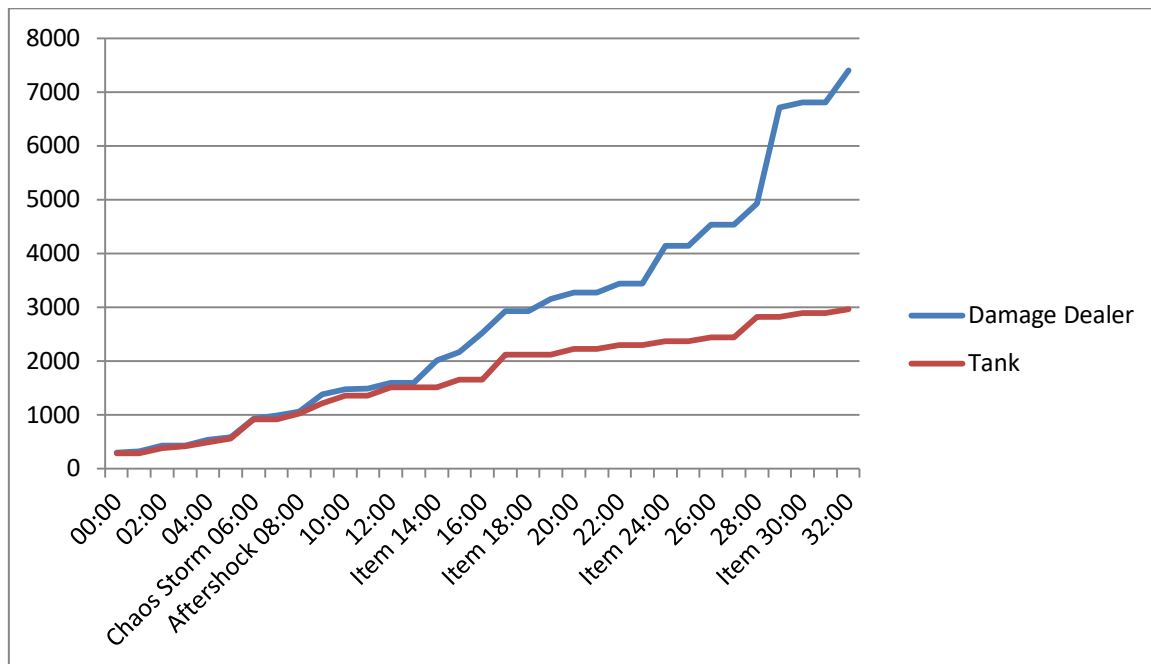
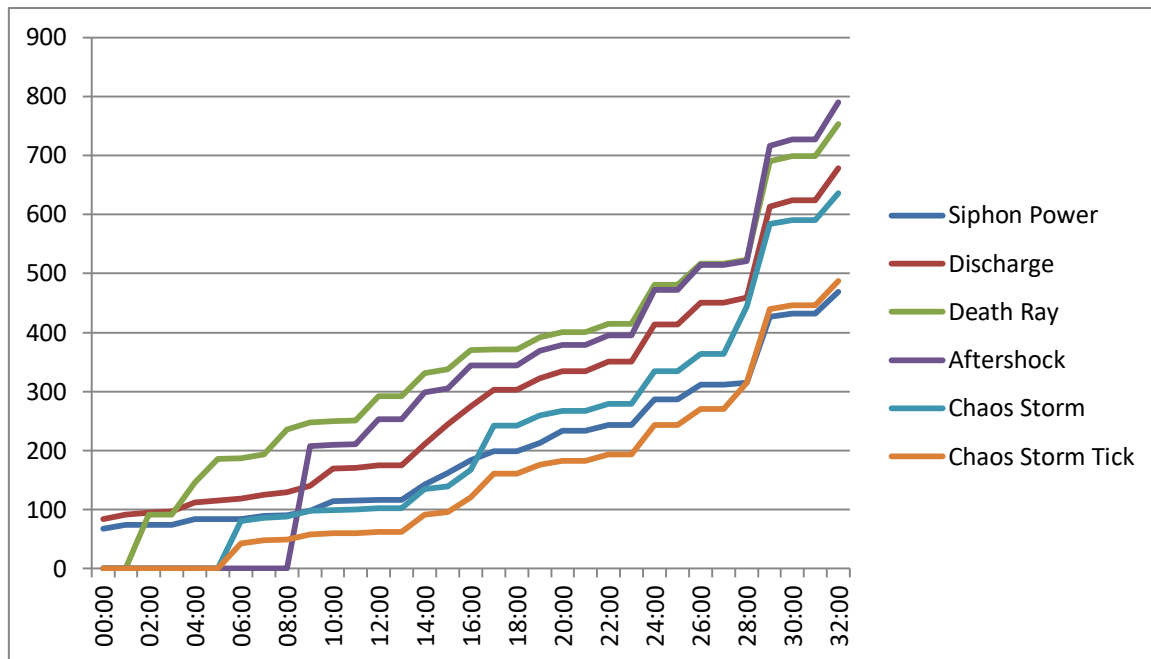
Siphon Power (5)
65 Mana
Total: 1021 [Q]
4s Cooldown

Viktor blasts an enemy unit, dealing 120 (+228) magic damage while granting Viktor a shield that absorbs up to 321 damage over the next 2.5 seconds.
Viktor's next basic attack deals 120(+314) bonus magic damage.
Augment - Turbocharge: Siphon Power's shield is increased by 60% and Viktor gains 30% Movement Speed for 2.5 seconds.

Click or press [Ctrl-Q] to level up

Cooldown	7.2 → 6.4
Mana Cost	45 → 50
Damage (Missile)	60 → 75
Damage (Attack)	20 → 45

Die folgenden Diagramme wurden mit Daten aus einer Partie League of Legends gefüttert. Dadurch wird sowohl das Leveln des Champions, wie auch das Kaufen von Items und somit der realistische Zuwachs an Ability Power gewährleistet.



Es sind also Power Spikes zu bestimmten Zeitpunkten (Minute 00:00 bis 32:00) im Spiel sichtbar. Diese sind entweder mit dem erstmaligen Leveln eines Skills, oder mit dem Erhalt eines wichtigen Items und durch das Wirken des Ability Power Scalings zu begründen. Würde Viktor anstelle von Ability Power in defensive Stats wie Armor, Magic Resistance oder Health Points investieren, so würde er nicht keinen Schaden machen, sondern auf sein Late Game Scaling verzichten.

Ein Modell mit Base Damage auf diese Art und Weise umgesetzt hat viele designtechnische Vorteile, da es dem Entwickler erlaubt, eine vertiefte Diversität für die Champions zu erstellen, die nicht nur mit Statuswerten wie Angriff und Verteidigung arbeiten, sondern auch mit der Dauer einer Partie, um einen dynamischeren Spielverlauf zu erschaffen. Ohne Base Damage gäbe es kaum einen Unterschied zwischen den einzelnen Spielphasen (Early, Mid, Late Game), geschweige denn zeitbasierten Stärken und Schwächen von Champions. Ausserdem erlaubt es dieses Modell dem Game Designer, einem Champion via schlechten Damage Scalings als Tank attraktiver zu gestalten.

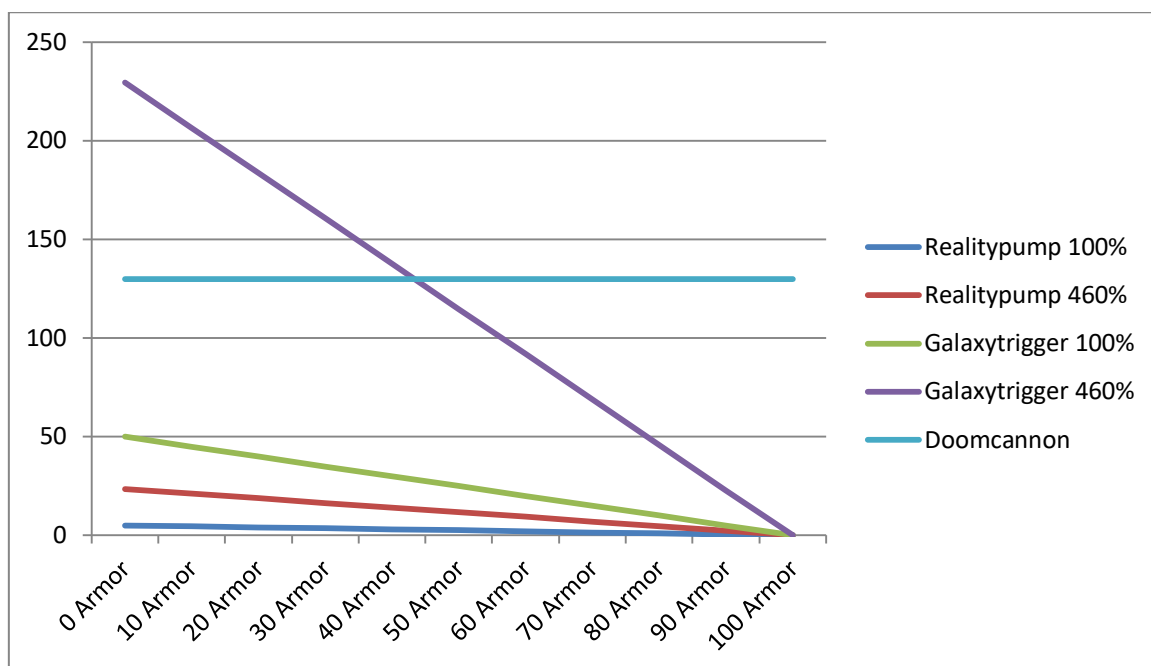
Game 5: Starbound

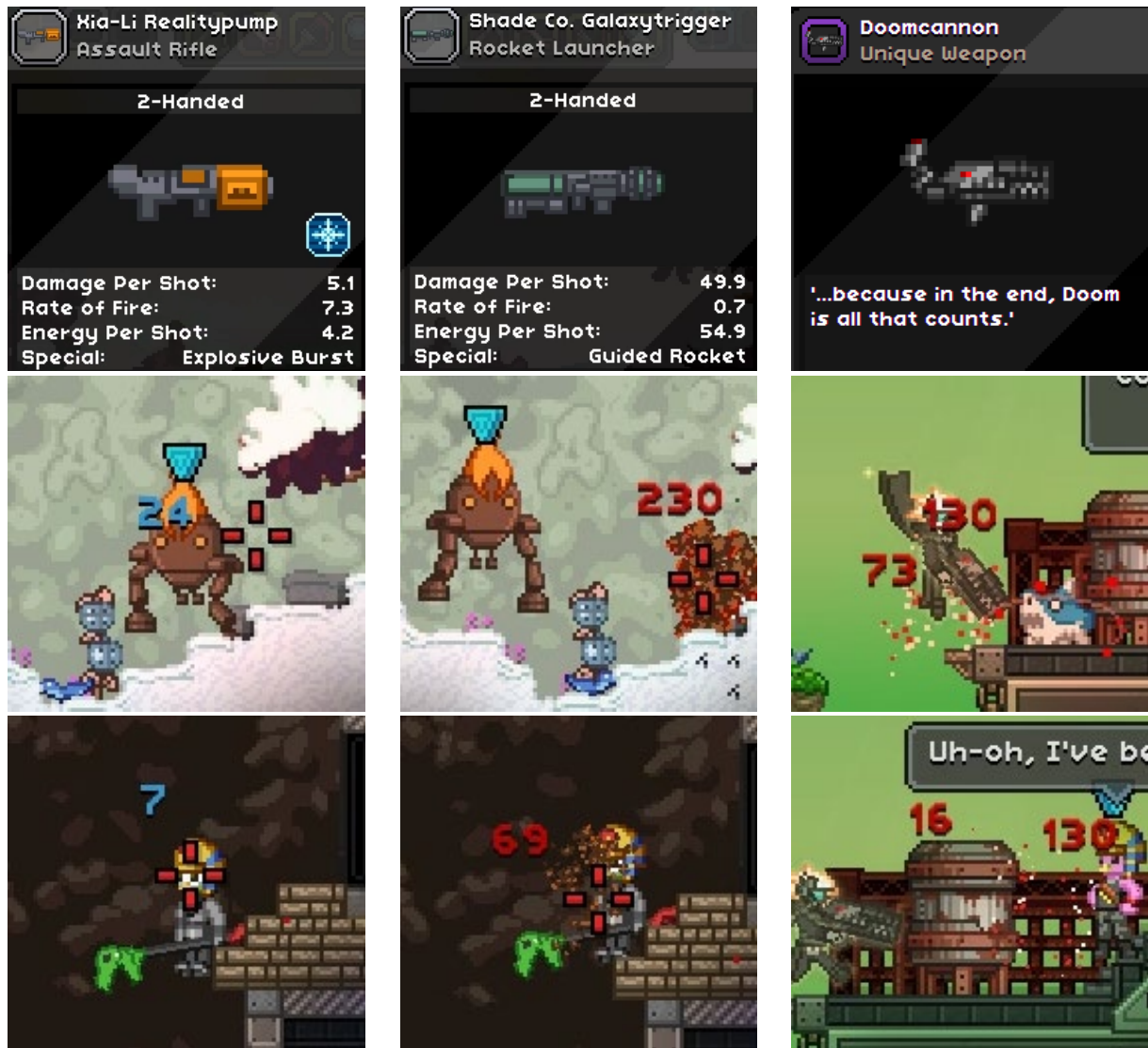
Genre: RPG (role-playing game) / Action Adventure / Sandbox

Ein weiteres Spiel mit einem prozentualen Scaling ist Starbound. Man spielt einen Charakter und reist von Planet zu Planet und sammelt Ressourcen, um neue Gegenstände wie Waffen, Rüstungen aber auch Werkstätten und rein ästhetische Objekte zu craften oder zu bergen. Dabei spielt der Combat eine Hauptrolle. Da das Schadenssystem simpel ist, aber sehr gut funktioniert, wird es in dieser Arbeit inklusive Rüstungswert analysiert, um weitere Schlüsse ziehen zu können. Die Formel zur Schadensberechnung lautet:

$$\text{Damage} = (\text{Weapon damage} * \text{Damage multiplier}) - \text{Armor\%}$$

Anstelle verschiedener Skills wird der Spieler dazu verleitet, verschiedene Waffen zu benutzen. Diese funktionieren unterschiedlich und verfügen über einen festen Waffenschaden, welcher mit dem Multiplikator der restlichen Ausrüstung (Standard 100%) skaliert. Hier der Vergleich zwischen drei Waffen auf einem Charakter mit 460% Schaden an unterschiedlichen Zielen mit unterschiedlichen Rüstungswerten:





Generell skalieren alle Waffen in Starbound gleich, was das Game einerseits sehr balanced macht, andererseits einen fast gleichgültigen Stat erschafft; die Armor. Theoretisch ist das Resultat dasselbe, ob nun ein Ziel über 50 statt 0 Armor oder doppelt so viel Leben verfügt. Es gibt nur zwei Gründe für die Existenz des Armor Values. Einer im PvP, wenn ein Spieler sich um einen festen Wert heilt, so ist es effizienter, wenn dieser möglichst wenig Health Points aber viel Rüstung hat, denn dadurch heilt man prozentual gesehen mehr von seinem effektiven Leben. Jedoch kann man den Rüstungswert als Spieler kaum beeinflussen, da die robuste Ausrüstung lediglich mehr Lebenspunkte gibt. Der andere Grund ist die Existenz der einzigartigen Waffe „Doomcannon“, welche stets 130 Damage verursacht. Es handelt sich hierbei um eine von hunderten Waffen mit einzigartigem Effekt. Die Doomcannon ist somit einerseits eher schwach gegen wilde Kreaturen ohne Rüstung, andererseits sehr stark gegen NPC's mit hoher Rüstung.

Es macht Spass, die Doomcannon effizient zu benutzen und es fördert das Einsetzen der richtigen Waffe in der richtigen Situation. Möglicherweise hätte man hier ein interessanteres System kreieren können, indem man die Schadensberechnungsformel, respektive die Art und Weise, wie der Armor Stat integriert ist, anders implementiert hätte. Mit Ausnahme dieses kleinen Mankos ist Starbound allerdings ein sehr gut ausbalanciertes Game in Bezug auf die Schadensberechnung und das Scaling einer jeden Waffe ... mit Ausnahme der Doomcannon natürlich.

Game 6: StarCraft II

Genre: RTS (real-time strategy)

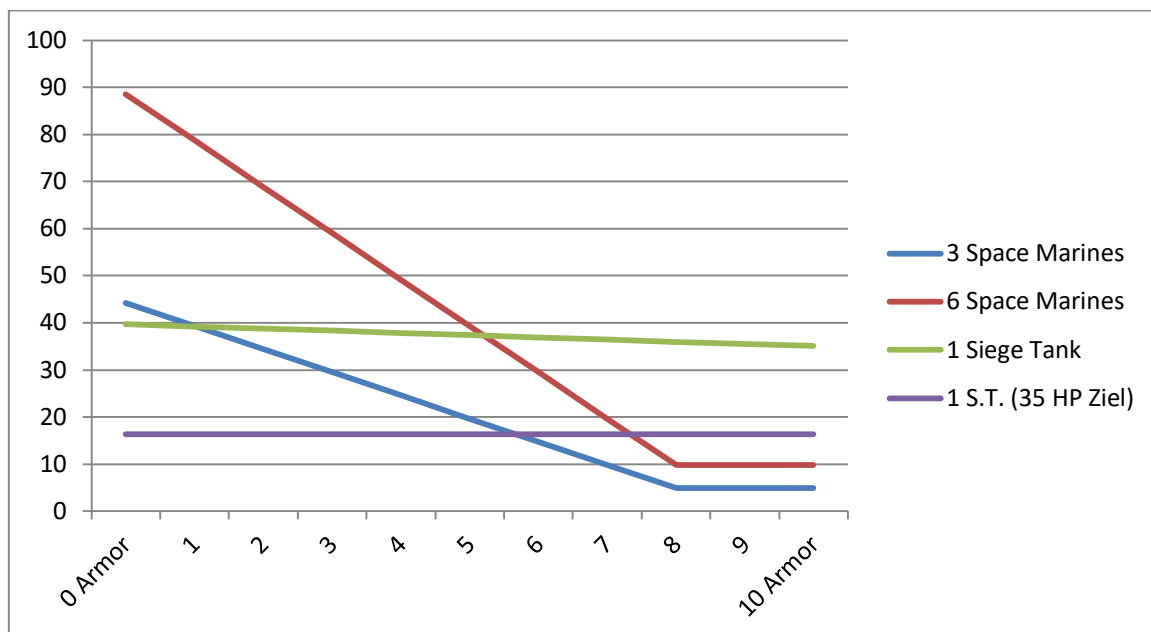
Bei StarCraft 2 handelt es sich um ein Aufbaustrategiespiel, bei dem es seine Basis und Armee aufzubauen und durch die bessere Strategie den Sieg zu erringen gilt. Die Einheiten können zwar im Spielverlauf einer Partie aufgewertet werden, doch werden diese durch die Aufwertungen des Gegners ungefähr ausgeglichen. Diese Arbeit befasst sich bei diesem Game allerdings mit der Interaktion zwischen den Damage, DPS (damage per second) und Armor Stats. Die Schadensberechnungsformel lautet hier:

$$\text{Damage} = \text{Unit Damage} - \text{Armor} \mid \text{Minimal Damage} = 1$$

Anders als bei Starbound reduziert der Armor Stat den Schaden um einen festen Wert. Dies spielt bei StarCraft II eine grosse Rolle, indem verschiedene Angriffe mehr oder weniger effektiv gegen ein Ziel sind. Es werden zwei verschiedene angreifende Einheiten gegen zwei verschiedene Ziele miteinander verglichen. Hierzu aber erst noch eine Rechnung zum Verständnis des Diagramms:

Angriffseinheit 1: Space Marine: $9 \text{ Damage} / 0.61\text{s} = 14.75 \text{ DPS}$
 Angriffseinheit 2: Siege Tank: $85 \text{ Damage} / 2.14\text{s} = 39.72 \text{ DPS}$
 Zieleinheit 1: Ultralisk: 7 Armor
 Zieleinheit 2: Zergling: 3 Armor

Für die Darstellung des Einflusses der Armor in diesem Schadensberechnungsmodell treten 3 Space Marines (44.26 DPS) gegen 1 Siege Tank (39.72 DPS) an, um ungefähr den gleichen Schaden zu simulieren und es treten 6 Space Marines gegen 1 Siege Tank an, um ungefähr auf die gleichen Ressourcenkosten zu kommen. Ausserdem tritt 1 Siege Tank gegen ein Ziel mit nur 35 HP an, um einen weiteren Nachteil sichtbar zu machen:



Dieses Diagramm repräsentiert lediglich einen stark isolierten Test, welcher nicht das Balancing von StarCraft II repräsentiert, sondern vielmehr die Wirkung einer festen Schadensreduktion mittels Rüstungswert aufzeigt. Im eigentlichen Game gibt es unzählige Faktoren, wie Reichweite, Bonus- und Flächenschaden, Ziele, Mobilität, Lebenspunkte etc., der Einheit, die das Balancing beeinflussen.

Es wird also sichtbar, dass ein einzelner starker Hit weniger von einer festen Damage Reduktion beeinflusst wird, als viele kleine Hits. Gleichzeitig kann es bei einem einzelnen starken Hit aber Nachteilhaft sein, indem er sein Ziel overkillt und somit ein Teil des DPS verloren geht.

Space Marine:



C-14 Gauss Rifle

Damage: 9

Range: 5

Weapon Speed: 0.61

Targets: Air and Ground

Siege Tank:



Crucio Shock Cannon

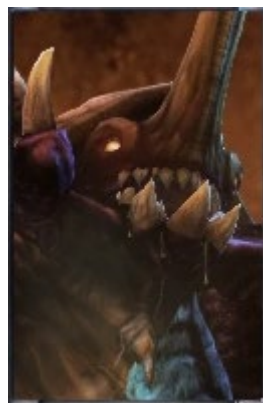
Damage: 52
vs Armored: 85

Range: 13

Weapon Speed: 2.14

Targets: Ground

Ultralisk (500 HP):



Zerg Ground Carapace

Armor: 7

Move Speed: 4.96

Zergling (35HP):



Zerg Ground Carapace

Armor: 3

Move Speed: 6.58

Game 7: Pokémon (Hauptspielereihe)

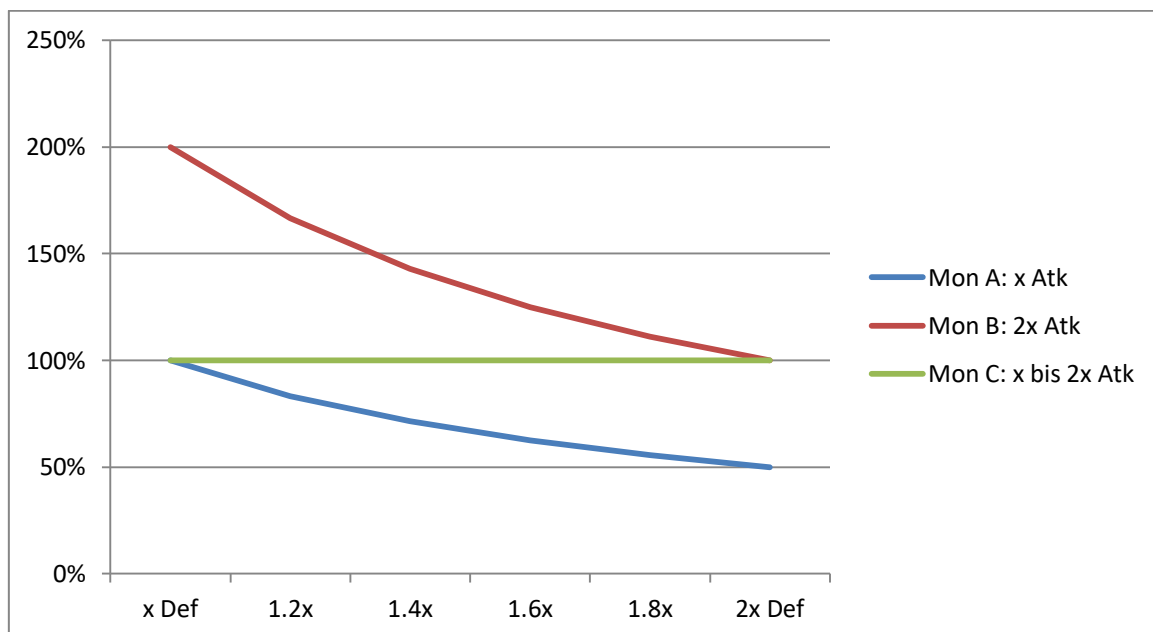
Genre: RPG (role-playing game) / TBS (turn-based strategy)

Die Pokémon Hauptspielereihe verfügt über ein rundenbasiertes Kampfsystem, bei welchem eine sehr komplexe Schadensberechnungsformel zum Zug kommt. In dieser Arbeit wird allerdings nur die Teilrechnung in Bezug auf den Angriffs- und Verteidigungswert analysiert, welcher als Modifikator für die gesamte Rechnung fungiert.

Ein Pokémon Kampf selbst ist quasi als Spielepartie anzusehen. Dabei können die einzelnen Stats eines Pokémon verstärkt oder geschwächt werden. Beziehen sich diese Veränderungen auf offensive / defensive Werte, so spricht man von einer positiven, beziehungsweise negativen Stufe auf einer Skala von -6 bis +6, wobei ±0 der Ausgangswert ist. Das Scaling dieser Stufen ist wie folgt:

-6	-5	-4	-3	-2	-1	±0	+1	+2	+3	+4	+5	+6
2/8	2/7	2/6	2/5	2/4	2/3	2/2	3/2	4/2	5/2	6/2	7/2	8/2
25%	28.6%	33.3%	40%	50%	66.7%	100%	150%	200%	250%	300%	350%	400%

Da die Pokémon untereinander über komplett unterschiedliche Stats verfügen, zum Beispiel Mon A 100 Atk und Mon B 200 Atk, ist es von höchster Wichtigkeit, dass diese positiven und negativen Stufen mit einem prozentualen System arbeiten. Würde es sich um einen fixen Wert wie zum Beispiel 20 Atk pro Stufe handeln, so würde das offensiv schwächere Mon A seinen Angriff um 20% erhöhen, während Mon B bloss einen 10%igen Bonus bekäme. Dieses System erlaubt ausserdem, dass jede positive Stufe mit der gleichen negativen Stufe (vice versa), welche auf dasselbe Pokémon wirkt, multipliziert werden kann und sie sich gegenseitig negieren. Ebenso erzielt man das gleiche Ergebnis, wenn zwei positive oder negative Stufen auf jeweilig ein angreifendes und verteidigendes Pokémon wirken. Sprich ist der verursachte Schaden eines +2 Atk, -2 Atk Pokémon = ±0 Atk und der verursachte Schaden eines +2 Atk Pokémon auf eines mit +2 Def ebenso. Dies kann aus folgendem Diagramm herausgelesen werden:



Erkenntnisse

Obwohl die Gesamtformeln eines Damage oder Stats Scalings in unzähligen Formen und Tiefen existiert, kann man von zwei Hauptmodellen als Überkategorien sprechen: Ein **Percentage Model**, welches sich rein prozentual als Modifikator auf einen bestimmten Stat bezieht und ein **Flat Model**, welches mittels festen Werten, beziehungsweise Base Damage arbeitet. Letzteres kann zusätzlich ebenfalls mit einem prozentualen oder Stat basierten Scaling arbeiten, um so ein **Hybrid Model** zu erschaffen, doch wird es in einem Spielverlauf nie so balanced, wie das rein prozentuale Modell sein.

Doch was heisst balanced in diesem Fall? Fakt ist, dass es sich hierbei um das anzustrebende Ziel des Designs eines Games handelt. Weder der Gamer, noch der Game Designer selbst sind ohne das Kenntnisse über dieses Ziel beurteilen, ob ein Spiel balanced ist oder nicht. Es gilt also erst das Ziel des Balancing Designs zu eruieren, um dann das richtige Modell finden und es dann auf das Game anzuwenden zu können, um die perfekte Balance zu kreieren.

Quellen- und Datenverzeichnis

Alle Quellen und Daten stammen vom 30.10.2019.

<https://wiki.guildwars2.com/wiki/Damage>

https://wiki.guildwars2.com/wiki/Dynamic_level_adjustment

https://rappelz.fandom.com/wiki/Lightning_Bolt

https://rappelz.fandom.com/wiki/Flame_Burn

<https://rappelz.fandom.com/wiki/Corruption>

https://rappelz.fandom.com/wiki/Corpse_Mine

<https://www.ign.com/faqs/2004/champions-of-norrath-realms-of-everquest-character-faq-492378>

<https://leagueoflegends.fandom.com/wiki/Damage>

<https://leagueoflegends.fandom.com/wiki/Viktor>

<https://starbounder.org/Combat>

<https://starbounder.org/Stats>

<https://starbounder.org/Weapons>

<https://starbounder.org/Armor>

<https://starbounder.org/Doomcannon>

https://liquipedia.net/starcraft2/Damage_per_second

<https://liquipedia.net/starcraft2/Armor>

[https://liquipedia.net/starcraft2/Marine_\(Legacy_of_the_Void\)](https://liquipedia.net/starcraft2/Marine_(Legacy_of_the_Void))

[https://liquipedia.net/starcraft2/Siege_Tank_\(Legacy_of_the_Void\)](https://liquipedia.net/starcraft2/Siege_Tank_(Legacy_of_the_Void))

[https://liquipedia.net/starcraft2/Ultralisk_\(Legacy_of_the_Void\)](https://liquipedia.net/starcraft2/Ultralisk_(Legacy_of_the_Void))

[https://liquipedia.net/starcraft2/Zergling_\(Legacy_of_the_Void\)](https://liquipedia.net/starcraft2/Zergling_(Legacy_of_the_Void))

<https://www.pokewiki.de/Statuswerte>

Alle Quellen und Daten, die nicht in den Links zu finden sind, stammen aus eigenem Research.