Institutionen för kommunikation och information
Examensarbete i medier: Dataspelsutveckling 30hp
C-nivå
Vårterminen 2008

Att balansera eller inte

En inblick i balanseringen av spelet Project Colosseum

Jonas Nilsson

Högskolan i Skövde
Institutionen för Kommunikation och Information
Medier
vt 2008

Att balansera eller inte – en inblick i balanseringen av spelet Project Colosseum

Examensrapport inlämnad av Jonas Nilsson till Högskolan i Skövde, för Kandidatexamen (B.A.) vid Institutionen för kommunikation och information. Arbetet har handletts av Anette Lundin.

9/5-08

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: _____

Högskolan i Skövde
Institutionen för Kommunikation och Information
Medier
vt 2008

Att balansera eller inte – en inblick i balanseringen av spelet Project Colosseum

Jonas Nilsson

Sammanfattning

Denna uppsats beskriver det praktiska arbetet bakom balanseringen av skadevärden gentemot de två fighting-stilarna *Mace* och *Dual-Wield* i spelet *Project Colosseum*.

Uppsatsen inleds med en beskrivning av bakgrunden till spelet, varefter de verktyg som använts i processen beskrivs i detalj.

Vidare presenteras praktiska exempel på uträkningar innan ett antal teorier rörande balansering, hämtade från facklitteratur presenteras.

Avslutningsvis förs en diskussion kring arbetets validitet, dess förhållande till de teorier som presenterats samt den egna kunskapsprocessen.

Nyckelord: Dataspel, balans, fighting, skadevärden

Högskolan i Skövde
Institutionen för Kommunikation och Information
Medier
vt 2008

Innehållsförteckning

1 Inledning

1.1 Bakgrund

Detta avsnitt syftar till att ge läsaren en inblick i bakgrunden till spelet ”*Project Colosseum*” samt en viss inblick kring den kontext/miljö arbetet skett i. En mer detaljerad beskrivning av spelet i sig går att finna under nästa avsnitt.

Project Colosseum är i grund och botten en fortsättning på ett tidigare projekt, som drevs under kursen *Spelprojekt II* på *Högskolan i Skövde* (officiell hemsida), VT-06. Under 10 veckor lyckades ett antal studenter skapa ett 3D fighting-spel [1] som då fick namnet *Colosseum*. Spelet fick god feedback från såväl lärare som studenter och medlemmarna i gruppen funderade på att arbeta vidare med produktionen.

Ett drygt år senare bestämde sig ”kärnan” i gruppen för att arbeta vidare på spelet, med förhoppningen att så småningom kunna lansera det kommersiellt. Nya medlemmar rekryterades (däribland jag), gruppen *Shortfuse Entertainment* och företaget *Gothia Science Park* (officiell hemsida) kontaktades.

Gothia Science Park (GSP) är en statligt finansierad institution som syftar till att hjälpa mindre företag vid etablering av verksamhet. GSP har en väl etablerad relation till högskolan och programmet för Dataspelsutveckling. Det nykomponerade teamet fick så småningom en plats i ”*The Game Incubator*” (officiell hemsida) och där är vi kvar än idag. Flytten till denna miljö innebar att tidigare nämnda förhoppningar förbyttes till en seriös målsättning, som i slutändan är tänkt att resultera i ett spel av så pass god kvalitet att det skall gå att lansera kommersiellt.

På GSP arbetar jag tillsammans med en annan designer (Fredrik Larsson) och vi är båda delaktiga i samtliga beslut som rör spelets design. Övriga medlemmar i gruppen (9 st.) har ständigt möjlighet att uttrycka sina åsikter när det gäller design men vår avdelning har ”avgörande röst” (samt det ansvar detta medför) på samtliga punkter som specifikt rör spelets design.

Det arbete som denna uppsats resulterat i har dock inte riktigt följt det ovan beskrivna arbetssättet. Jag har fått mycket hjälp med att bolla idéer, diskutera förslag samt testa implementerade[2] element men allt ”faktiskt” arbete (beräkningar, införande av värden, etc.) som gäller design är resultat av mina egna, individuella ansträngningar. Det är, trots detta, viktigt att påpeka att mina ansträngningar inte skulle vara möjliga utan mina kära kollegor och ett stort tack riktas därför till samtliga ”anställda” på *Shortfuse Entertainment*!

[1] Ett spel som i korthet går ut på att besegra sin motståndare genom att slåss mot denne.
[2] Att se till att idéer förverkligas, exempelvis att design omsätts till programkod.

1.2 Project Colosseum

Det känns lämpligt att inleda detta kapitel med ett utdrag ur det high-concept[3] som skapades för det ursprungliga Colosseum:

"*In the desert of Dasht-e-kavir, a massive arena stands tall. The city in which the arena lies is ruthless, governed by the laws of an ancient code written down by the great Babylonian king - Hammurabi. The code states that anyone who claims entrance to the city must prove their worth in the arena. This particular rule, though its importance has diminished over time, still lays the basis of the city's rule. Fame, wealth and power - all can be gained and lost in the arena.*"

Tanken med detta citat är att läsaren (förhoppningsvis) lyckas skapa någon form av mental bild av den stämning *Project Colosseum* försöker eftersträva. Själva ”essensen” av spelupplevelsen bygger alltså på att kämpa mot andra gladiatorer i en arena, med livet som insats.

Project Colosseum bör betraktas som en fristående uppföljare till det ursprungliga Colosseum men båda spelen bygger ändå på samma grundläggande tanke. *Projekt Colosseum* är ett spel av typen 3D-fighting och kan liknas vid välkända titlar, såsom *Soul Calibur III* (Namco, 2005).

I korthet går spelet ut på att med hjälp av olika offensiva och defensiva manövrar besegra sin motståndare i strid. Spelaren antar rollen som gladiator i en arena, där denne kämpar mot en eller flera motståndare. Den del av spelet som behandlas i denna rapport består av speltypen *Deathmatch* [4].

I dess nuvarande form innehåller *Project Colosseum* två unika ”stilar” som spelaren kan välja mellan. Dessa har benämningarna *Mace* och *Dual-Wield*. Det är arbetet med balansering av dessa stilar som ligger till grund för uppsatsen och en kort introduktion är därför på sin plats.

Den generella tanken bakom designen av dessa stilar kan enklast beskrivas på följande sätt:

1.2.1 Mace

- *Ett stort, tungt vapen.*
- *Långsamma attacker.*
- *Stor räckvidd.*
- *Mycket skada (per slag).*

1.2.2 Dual-Wield

- *Två mindre, snabba vapen.*
- *Snabba attacker.*
- *Kort räckvidd.*
- *Lite skada (per slag).*

[3] Ett dokument som beskriver spelets idé, gameplay och story på ett generellt sätt.
[4] Ett spel-läge som går ut på att döda sina motståndare flest gånger, utefter angivna villkor. Den som först uppnår den summa som motsvarar villkoret (exempelvis 10 ”kills”) vinner.

Dessa faktorer har varit något av ”ledstjärnor” vid den designmässiga utvecklingen. De tydliga kontrasterna i ovannämnda avseenden torde innebära två vitt skilda sätt att spela på och detta har från början varit en grundläggande tanke för spelet. I *Project Colosseum* är det viktigt att kunna ta sig an strid på flera olika sätt, beroende på vilken stil man väljer för sin gladiator. Den grundläggande tanken är kort och gott att en skicklig spelare skall ha (i princip) lika goda möjligheter att vinna med båda stilarna, förutsatt att denne behärskar dem båda lika bra.

Detta påstående är naturligtvis svårt att bevisa och det bör nämnas att denna uppsats inte är av kvantitativ natur, vilket i sig innebär att påståendet inte kommer att ”kontrolleras” med hjälp av diverse mätningar och tester.

1.3 Syfte

Syftet med denna uppsats är att ge läsaren en inblick i det arbete jag utfört på spelet *Project Colosseum* med avseende på balansering av två ”stilar” (*Mace* och *Dual-Wield*) gentemot varandra. Vidare syftar även uppsatsen till att presentera tankar om speldesign från etablerade verk inom branschen samt diskutera hur det utförda arbetet förhåller sig till dessa. Avslutningsvis diskuteras de lärdomar som dragits samt reflektioner som uppstått under arbetets gång.

1.4 Avgränsning

Det är viktigt att poängtera att syftet med denna uppsats **inte** är att utvärdera det utförda arbetet på ett absolut sätt. Jag har inget intresse av att utvärdera mitt arbete utefter någon form av ”mall”, som skulle avgöra vad som är explicit ”bra” eller ”dåligt”. Detta vill jag undvika eftersom jag anser det vara mycket svårt att avgöra vad som faktiskt är ”bra” eller ”dåligt”. Att försöka sig på att definiera dessa termer (inom detta område) anser jag kräver en egen uppsats i sig, varför jag väljer att bortse från *absoluta* gränsdragningar.

Detta till trots kommer ändå en form av personlig utvärdering av det utförda arbetet ske. Genom att redovisa uträkningar och med den diskussion som förs i slutet av uppsatsen hoppas jag kunna besvara frågan:

”Vilka designmässiga beslut avseende balansering av de två stilarna Dual-Wield och Mace har tagits och hur fungerar de?”

I diskussionen kommer även detta resultat att analyseras utifrån de teorier som tas upp. Utifrån dessa kommer arbetet säkerligen att hamna i kontrast till andra åsikter men i denna uppsats gör jag valet att inte tolka detta som något explicit negativt eller positivt.

Vidare har beskrivningar av vissa element (exempelvis spelets kontroller) utelämnats för att minska omfattningen på arbetet. Den diskussion som i slutändan förs berör så pass abstrakta tankar och funderingar att praktisk kunskap kring hur detta realiseras i spelet inte anses vara nödvändig för läsarens förståelse av balanseringsarbetet.

2 Attacktyper

Detta avsnitt syftar till att ge läsaren en förståelse för bakgrunden till de olika typer av attacker som existerar i *Project Colosseum*. Avsnittet har stor vikt för läsarens förståelse av kommande avsnitt, då den fakta som presenteras nedan ligger till grund för det praktiska arbetet.

Förutom en generell överblick av de olika attacktypernas funktionalitet presenteras även de bakomliggande designmässiga tankarna kring deras skadevärden.

2.1 Skada

Project Colosseum går i grund och botten ut på att åsamka sina motståndare mer skada än de åsamkar dig. Detta kan ske på ett antal olika sätt, alla med en (tänkt) individuell ”funktion”.

De värden som anges nedan gäller i förta hand för stilen *Mace*, vars attacker fungerar som riktvärde för balansering av *Dual-Wield*. Detta arbete beskrivs i större detalj i ett kommande kapitel (Balansarbete – s. 18). Notera att dessa värden är spelets ursprungliga värden, som sedan kommer att revideras vid balansering.

När termen ”totalen” nämns nedanför syftar detta till varje spelares totala mängd hälsa. Denna variabel går att modifiera fritt men har antagits ha det konstanta värdet **700** ”enheter” för arbetet i denna uppsats.

De fyra olika sätt skada kan göras på i *Project Colosseum* är:

- *Basic attacks*
- *Combos*
- *Charge attacks*
- *Rage attacks*

2.2 Basic attacks

Att utföra en *Basic attack* är det enklaste men minst effektiva sättet att skada sin motståndare på. Det finns tre olika sorters *Basic attack*s: *left*, *right* och *forward*.

Basic attacks är designade att göra väldigt lite skada. Grundtanken är att det skall vara oerhört ineffektivt att döda en motståndare enbart genom att använda *Basic attacks*. Det totala skadevärdet för en attack av denna typ har därför i detta arbete bestämts motsvara **2,5 %** av totalen. *Basic attacks* inleder combos (sen nedan) men deras skadevärden är alltid de samma, oavsett combotyp.

Notera att samtliga procentuella värden som satts alltså bör ses som designmässiga beslut som tagits under arbetets gång.

2.2.1 Swings

En viktig aspekt att ta hänsyn till vid utförandet av attacker är ”swings”. Swings kräver mer avancerad input än en ”vanlig” attack men dubblerar skadevärdet. Noterbart är även att swings endast kan utföras på attacker av typen *left* och *right*.

Det är alltså möjligt att dubbla skadevärdet på samtliga vänster- eller högerattacker i spelet, inte bara *Basic attack*s. Det är med andra ord möjligt (i teorin) att dubblera skadan på en hel combo (se nedan) genom att utföra swings på sina attacker.

Swings är en viktig faktor att ta hänsyn till vid specificering av skadevärden eftersom de i praktiken innebär att en attack har möjlighet att göra dubbelt så mycket skada som anges i *Toolkit* (se kapitlet Verktyg – s. 9). Detta innebär att samtliga värden för *höger*- och *vänster*attacker i *Toolkit* måste halveras från sitt ursprungliga riktvärde för att inte få för stora skadevärden.

Sker inte denna numeriska halvering stämmer inte det ”reella” skadevärdet överrens med designen. En *Basic attack* av typen *left* eller *right* som inte halveras har ett potentiellt skadevärde på **5 %** av totalen (2,5 % x 2) eftersom dessa alltså kan utföras som swings – ett beslut som fattades i den ursprungliga designen av Project Colosseum.

I slutändan måste därför skadevärdet hos **samtliga** attacker (inte bara *Basic attacks*) av typen *left* eller *right* halveras innan deras värden förs in i *Toolkit*.

2.3 Combos

En combo är en serie av sammanlänkade attacker (animationer) som tillsammans resulterar i att ett flertal unika animationer spelas upp. Varje ”steg” i combon har en särskild animation kopplad till sig, som spelas upp ifall spelaren ger den input[5] som motsvarar denne. Gemensamt för varje combo är att de måste inledas med en *Basic attack* samt att den input de kräver är förbestämd och inte går att ändra på.

Varje combo följer en progressivt expanderande trädstruktur, vilket innebär att spelaren kontinuerligt får visuell feedback på ”var i combo:n de befinner sig” – representerat av animationer. Mer information om hur denna trädstruktur fungerar finns under kapitlet Verktyg (s. 9).

Ifall spelaren ger input som inte överrensstämmer med den förbestämda input:en för en combo kommer denna att avslutas så fort animationen för den sista ”korrekta” inputen spelats upp. Combos kräver därför en viss skicklighet av spelaren och tanken är att ”belöna” denna skicklighet genom att skadevärdet för attacker ökar successivt, för varje ”steg” i combo:n. För att göra mesta möjliga mängd skada är det därför viktigt för spelaren att behärska utförandet av combos.

[5] Input är hur en handling hos en användare omvandlas till data som programvaran kan förstå, exempelvis att ett knapptryck kan resultera i att en karaktär hoppar i ett spel.

2.3.1 Combotyper

Det finns fyra olika ”typer” av combos, alla med ett (tänkt) eget ”syfte”. Nedan följer en kort beskrivning av varje combotyp:

- **Light combos**
  - *Enkel input.*
  - *Mindre skillnader i skada för varje animation/”steg”.*
  - *Har ett sammanlagt skadevärde motsvarande **15 %** av totalen.*

- **Heavy combos**
  - *Svårare input.*
  - *Större skillnader i skada per animation/steg (större skadeökning på de sista animationerna).*
  - *Har ett sammanlagt skadevärde motsvarande **25 %** av totalen.*

- **Advanced combos**
  - *Svårast input.*
  - *Lite skada på vanliga attacker, förutsättning för Special attacks (se nedan).*
  - *Har ett sammanlagt skadevärde motsvarande **40 %** av totalen, fördelat på två ”etapper”:*
    - *”Vanliga” slag i combon = **10 %***
    - *Special attack slag i combon = **30 %***

- **Area of Effect (AOE) combos**
  - *Enkel input.*
  - *Täcker stor yta, gör lite skada.*
  - *Har ett sammanlagt skadevärde motsvarande **12,5 %** av totalen.*

Meningen med att ha ett särskilt syfte bakom varje combotyp är att premiera varierat användande av combos. Varje combo behöver därför inte bara sättas i relation till sin motsvarighet för övriga stilar utan även internt (gentemot övriga combotyper).

Ett bra exempel på detta är jämförelsen mellan *Advanced combos* och *Light combos*. *Advanced combos* kan potentiellt göra mer skada (på grund av sin *Special attack* – se nedan) men en skicklig spelare kan motverka denna skada genom att undvika just denna attack/animation.

Om denne sedan attackerar sin motståndare med en *Light combo* är det potentiella skadevärdet **5 %** *högre* (**15 %** > **10 %** - *Special attack* har misslyckats). En *Advanced combo* kan således göra betydligt mer skada men bara under förutsättning att dess *Special attack* aktiveras.

Den generella tanken är alltså att ”optimalt” spelande består av att välja combotyp utifrån sin motståndares handlingar och vilka combos de väljer att använda. Det skall med andra ord inte vara möjligt att besegra en eftertänksam spelare genom att utföra samma combo om och om igen, eftersom denne då kan motverka dina attacker genom att välja en annan combotyp (som ”motverkar” just den typen av combo som för tillfället används).

2.3.2 Special attacks

*Special attack*s är särskilda animationer som återfinns i *Advanced combos*. Dessa animationer är unika på det sättet att spelaren inte har direkt kontroll över deras uppspelning samt att skadevärdet på dessa attacker är väldigt högt.

I den bilaga som visar input för samtliga combos representeras *Special attack*s som en attack med parentes omkring sig. Detta eftersom det inte är garanterat att denna kommer att spelas upp, även ifall spelaren anger korrekt input för combon. Anledningen till detta är att en *Special attack* i sig är väldigt kraftfull och skall fungera som en sorts ”belöning för skickligt spelande”.

För att en *Special attack* skall spelas upp krävs att den sista attacken är kopplad till någon form av attack-input (sista attacken innan parentesen) träffar motståndaren. Denna attack kallas ”test-blow” eftersom det är denna som avgör ifall *Special attack* animationen kommer att spelas upp eller inte.

Om en *Special attack* ”aktiveras” innebär detta att en animation som spelaren inte angett input för själv spelas upp (för spelaren), samt att deras motståndare blir temporärt ”satt ur spel” (en särskild animation spelas upp för motståndaren och input slutar temporärt att fungera – de kan inte försvara sig mot attacken) Den generella tanken med en *Special attack* är alltså att de innebär en form av risktagande. Lyckas man träffa sin motståndare med combons test-blow och aktivera sin *Special attack* delas en stor mängd skada ut.

Special attacks har en sammanlagd skademängd motsvarande **30 %** av totalen.

Skulle motståndaren lyckas undvika själva *Special attack*en har man i princip ”gått miste om utebliven skada” eftersom det sammanlagda skadevärdet fram till själva *Special attack:en* är lågt jämfört med övriga combotyper (endast **10 %** av totalen).

2.4 Charge attacks

Det finns två sorters *Charge attacks*:

- **Charge left/right**
  - *Area of Effect (AOE) attack*
    - *Gör lite skada.*
    - *Täcker stor yta.*
- **Charge forward**
  - *Daze attack*
    - *Gör lite skada.*
    - *Motståndare blir temporärt ”satt ur spel”.*

Charge attacks kan utföras när som helst under spelets gång och avbryter en pågående combo när input har registrerats av spelet. Notera även att vänster och höger attack måste utföras som swings för att input för en *Charge attack* skall registreras.

Båda typerna av *Charge attacks* har ett sammanlagt skadevärde motsvarande **7,5 %** av totalen.

2.5 Rage attacks

En *Rage attack* aktiveras enkelt genom ett knapptryck men är begränsad på det sätt att den har en aktiveringskostnad kopplade till sig (de kräver *Rage*).

Kortfattat om *Rage*:

- *Tilldelas när spelare **ger** eller **tar emot** skada.*
    - *Ökar mer om man **tar emot** skada.*

Både *Mace* och *Dual-Wield* har bara en *Rage attack* tillgänglig. *Rage attack*s är speciella eftersom de inte alltid kan aktiveras (de kräver *Rage*), inte kräver någon särskild input samt gör stor mängd skada om de träffar motståndaren.

Rage attacks har ett sammanlagt skadevärde motsvarande **25 %** av totalen.

2.6 Tabell

Nedan presenteras en tabell (Tabell 1) där procentuella värden och faktiska skadevärden för varje attacktyp (för stilen *Mace*) samlats. Läsaren bör notera att de procentuella värden som inte resulterat i heltal avrundats uppåt. Skadevärden för *Dual-Wield* redovisas inte i detta skede, eftersom de i slutändan beror på nedanstående värden.

Tabell 1 – procentuella värden och faktiska skadevärden för olika attacktyper (*Mace*)

Attacktyp	Sammanlagt värde av totalen - procentuellt	Reellt skadevärde - enheter
Left attack	**2,5 %**	**9** (18/2, p.g.a *swings*)
Right attack	**2,5 %**	**9** (18/2, p.g.a *swings*)
Forward attack	**2,5 %**	**18**
Light combo	**15 %**	**105**
Heavy combo	**25 %**	**175**
Advanced combo (”vanliga” slag)	**10 %**	**70**
Advanced combo (Special attack)	**30 %**	**210**
Area of Effect combo	**12,5 %**	**88**
Charge attack	**7,5 %**	**53**
Rage attack	**25 %**	**175**

3 Verktyg

Detta avsnitt syftar till att förklara det verktyg som primärt använts i arbetet med *Project Colosseum*. De punkter som är av vikt för vidare förståelse av arbetet kommer att förklaras i större detalj men läsaren bör komma ihåg att de övriga aspekter av programvaran *inte* kommer att presenteras (eftersom de inte bedöms vara av vikt för läsarens vidare förståelse).

3.1 Toolkit

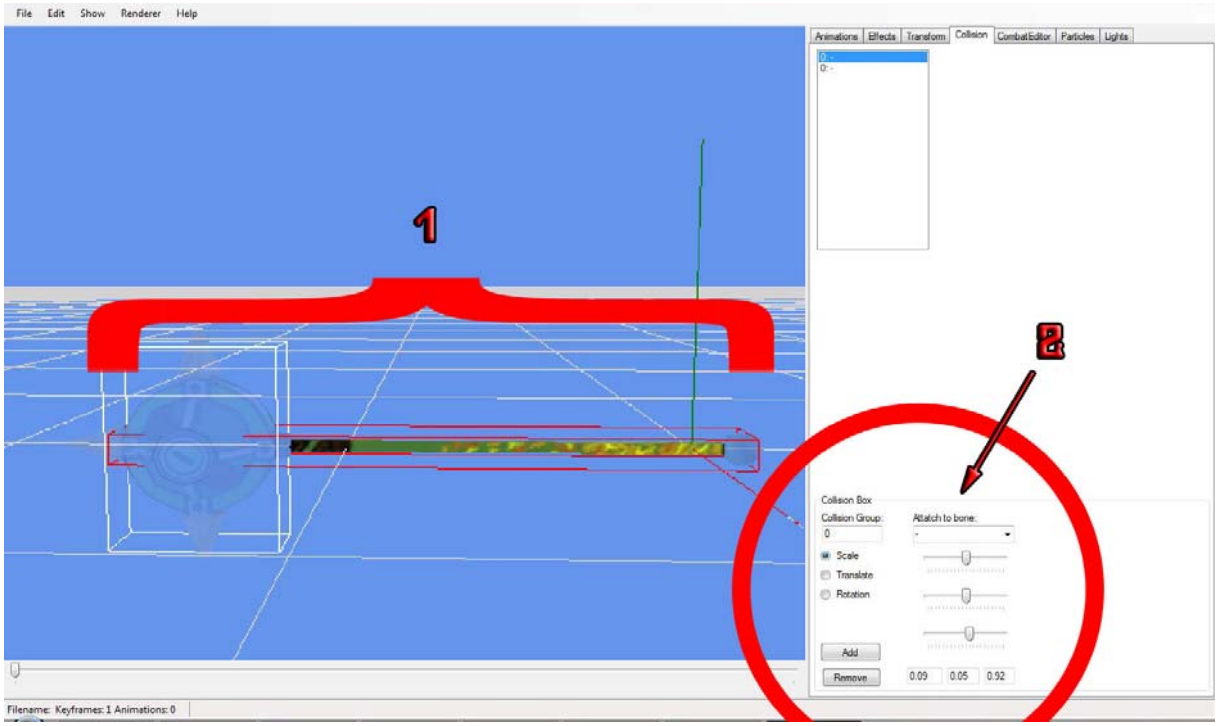
Toolkit (arbetsnamn) är en egenutvecklad programvara tillverkad av gruppens programmerare. Programmet har flera olika användningsområden och är till stor hjälp vid arbete med visuella detaljer, såsom animationer.

I programmet finns ett antal olika flikar för användaren att välja mellan. Relevanta för denna uppsats är flikarna *Collision* och *CombatEditor*.

3.1.1 Collision

För att en attack skall kunna åsamka en spelare skada krävs naturligtvis att vapnet träffar personen i fråga. I *Project Colosseum* måste den exakta tidpunkten för när detta skall ske specificeras för varje animation.

Varje föremål i spelet har en så kallad ”kollisionsbox” (se *Figur 1*) som omringar dem. När någon del av en sådan box träffar en annan (box) uppstår en kollision. I *Project Colosseum* kan en box vara **av** eller **på** under ett visst intervall. Ifall en kollision mellan olika boxar sker men variabeln för kollision är satt till **av** innebär detta att spelmotorn[6] ignorerar denna händelse och en ”faktisk” kollision inte sker (i spelvärlden).



Figur 1: Bild av ett vapen och dess kollisionsboxar.

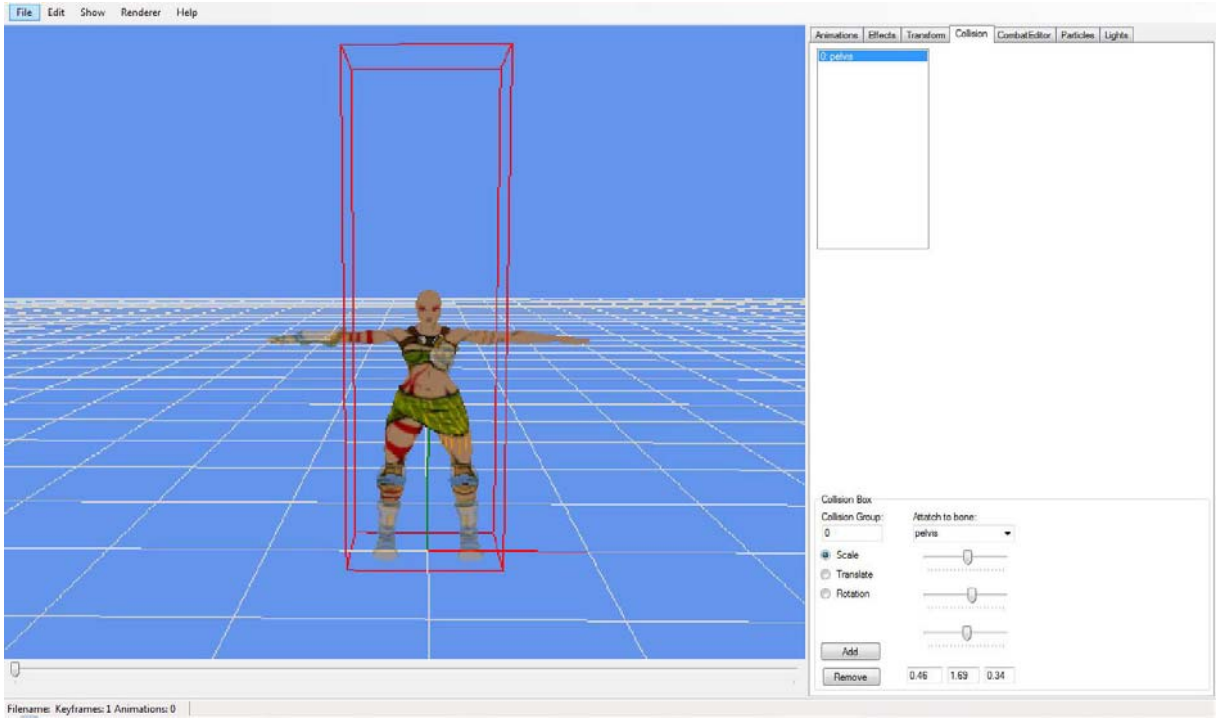
[6] ”Hjärnan” bakom spelet. Spelmotorn består av all den programmeringskod som behövs för att spelet skall fungera.

I verktyget är det möjligt att manipulera dessa kollisionsboxar genom att ändra deras placering kring vapnet samt öka/minska deras storlek. Med hjälp av dessa ändringar är det alltså möjligt att **öka** ett vapens ”träffsäkerhet” (*större* box) eller **minska** den (*mindre* box). På bilden ovan (*Figur 1*) har användaren markerat den ena kollisionsboxen (**1**) och har därefter möjlighet att påverka dennes storlek eller placering (**2**).

Värt att notera är att kollision endast kan uppstå en gång per vapen och animation (undantaget *Dual-Wield* – se nedan). I praktiken innebär detta att när en kollision väl skett så avaktiveras kollisionen för återstoden av animationen.

För att motverka detta och möjliggöra flertalet kollisioner per attack/combo är det möjligt att ”dela upp” en längre animation i flera, kortare animationer. Denna process (”splits”) samt processen att aktivera/avaktivera kollision beskrivs i större detalj under ett kommande avsnitt (Splits – s. 15).

I spelet är samtliga vapens kollisionsboxar avaktiverade till dess att annat sägs. Det samma gäller dock inte för boxar på karaktärer (se *Figur 2*), eftersom dessa måste kunna kollidera med varandra utan att en attack skett (annars kan man exempelvis ”springa igenom” andra spelare). Anledningen till att kollisionsboxar på vapen inte aktiveras förrän vid specificerade tillfällen är för att omöjliggöra att ”konstiga” situationer uppstår.



Figur 2: Bild av en karaktär och dennes kollisionsboxar.

Ifall kollision på vapen ständigt vore aktivt skulle det exempelvis vara möjligt att göra skada på en motståndare när som helst som ett vapen rör vid dem, oavsett om en attack utförts eller inte. Detta skulle exempelvis betyda att en springande karaktär som håller sitt vapen vid sidan och ”snuddar” vid en motståndare delar ut skada till denne. Själva essensen i ett fighting-spel försvinner om varje vapen har en ”obegränsad potentiell dödlighet”, det känns dessutom väldigt ologiskt – det skall inte göra ont om man rör vid motståndare, bara om man slår dem!

En viktig skillnad mellan stilarna i spelet är att *Dual-Wield* har möjlighet till 2 kollisioner per animation (1 per svärd) medan *Mace* bara kan kollidera en gång per animation. Detta har stor vikt för balansarbetet eftersom det i praktiken innebär att skadan per slag måste halveras för *Dual-Wield*.

En kort sammanfattning av kollisioners betydelse:

- Skada kan inte delas ut om inte kollision mellan ”vapenbox” och ”människobox” sker.
    - *Kollision sker när två ”boxar” rör vid varandra.*
- Kollision kan aktiveras eller avaktiveras.
    - *Karaktärer har ständigt **aktiverad** kollision.*
    - *Vapen har ständigt **avaktiverad** kollision som måste/kan aktiveras vid behov.*
- *Dual-Wield* kan kollidera 2 gånger per animation, *Mace* bara 1.

Nästa avsnitt behandlar i detalj den andra delen av *Toolkit* som bör förklaras: *CombatEditor* (arbetsnamn).

3.1.2 CombatEditor

Denna del av *Toolkit* har haft störst inverkan på balansarbetet då det är här som samtliga numeriska värden förs in, för att sedan användas av spelet under körning.

För att kunna utveckla och vidare förklara mer abstrakta aspekter av varje attack/combo bör strukturen bakom en animation i sig beskrivas – dess key-frames.

Key-frames

När input som motsvarar en attack ges tolkar spelmotorn detta och spelar upp en animation, beroende på vilken sorts input som angivits (*left*, *right* eller *forward*).

En animation i sin tur består av ett antal olika key-frames vilka kan ses som animationens ”byggstenar”. Varje key-frame motsvarar en viss rörelse och när samtliga key-frames spelats upp har detta resulterat i en längre rörelse (som då alltså i själva verket består av flera mindre rörelser som ”lagts ihop”).

Varje key-frame motsvarar i sin tur en viss tidslängd, vilket innebär att den faktiska tiden som krävs för att spela upp animationen beror på det totala antalet key-frames.

Den totala tiden bestäms vidare i Project Colosseum av en variabel [7] som heter *Rules.Game.ANIM_SPEED_FPS*. Detta värde kan lätt ändras i en textfil och avgör hur många key-frames systemet spelar upp per sekund. I *Project Colosseum* används i nuläget värdet 25, vilket innebär att en animation som består av 25 key-frames tar 1 sekund att spela upp.

Vidare har varje animation 20 ”avslutande” key-frames där det inte sker särskilt mycket rörelse. Under denna tid (en dryg sekund – se ovan) är det möjligt för användaren att ge input som motsvarar nästkommande animation i combon. Närhelst ”korrekt” input angivits avbryts animationen och nästa animation/steg i combon spelas upp. Syftet med dessa key-frames är att förenkla utförandet av combos – användaren har en sekund på sig efter varje input att ange nästa input men kan avbryta denna paus tidigare (förutsatt att de anger ”korrekt” input).

[7] En variabel är ett värde som inte är konstant.

Effektiva key-frames

Ovan har det totala antalet key-frames för en animation beskrivits. Vid arbetet med balansering i *Project Colosseum* tas dock bara hänsyn till de key-frames som spelas upp innan de ”avslutande” key-frames:en i varje animation. Dessa key-frames benämns som ”effektiva” och i praktiken motsvarar de alltså antalet key-frames som **måste** spelas upp per animation (går ej att avbryta p.g.a ”korrekt” input).

Antalet effektiva key-frames har stor betydelse för specificering av skadevärde på en attack/animation i förhållande till två områden:

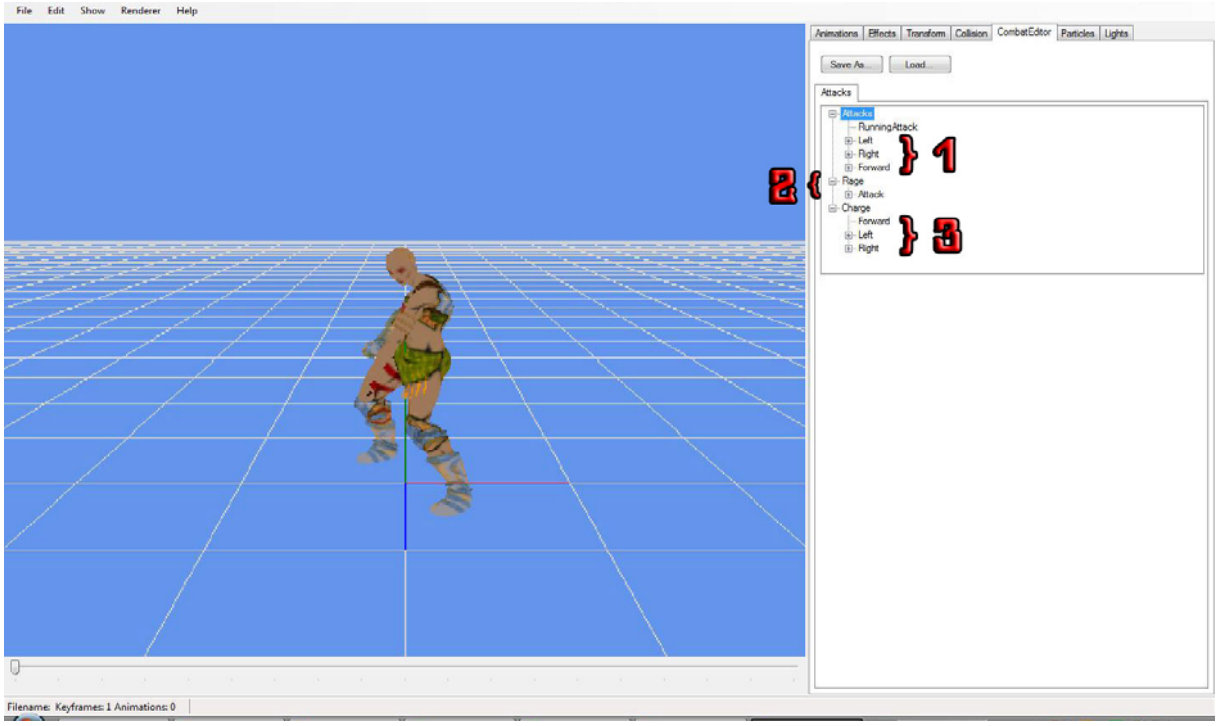
1. ”**Interna**” combos – Combos av samma typ inom **samma** stil (exempelvis två *Heavy Combos* för *Mace*).
2. ”**Externa**” combos – Combos av samma typ inom en **annan** stil (exempelvis *Heavy Combo* för *Mace – Heavy Combo* för *Dual-Wield*).

Skadevärdet på varje attack/animation i en combo måste alltså inte bara balanseras gentemot combos av samma typ för andra stilar utan även internt. Detta eftersom det inte skall finnas en combo som är ”bättre” än någon annan (har ett högre sammanlagt skadevärde) – för att främja ett varierat användande.

En kort sammanfattning av key-frames betydelse:

- Varje attack presenteras visuellt för spelaren med hjälp av en animation.
- En animation består av ett antal key-frames.
    - *Antalet key-frames avgör hur lång faktisk tid det tar att spela upp animationen.*
    - *”Effektiv” key-frame = Key-frame som **måste** spelas upp innan nästkommande animation kan påbörjas.*
        - *Viktig variabel att ta hänsyn till vid balansarbetet.*
            - *Internt – Combos jämförs **inom** stilen.*
            - *Externt – Combos jämförs **mellan** stilarna.*

För att läsaren skall kunna förstå övriga användningsområden för *CombatEditor* presenteras nu en bild som kommer att användas som utgångspunkt för vidare förklaring och utveckling av specifika funktioner (*Figur 3*).

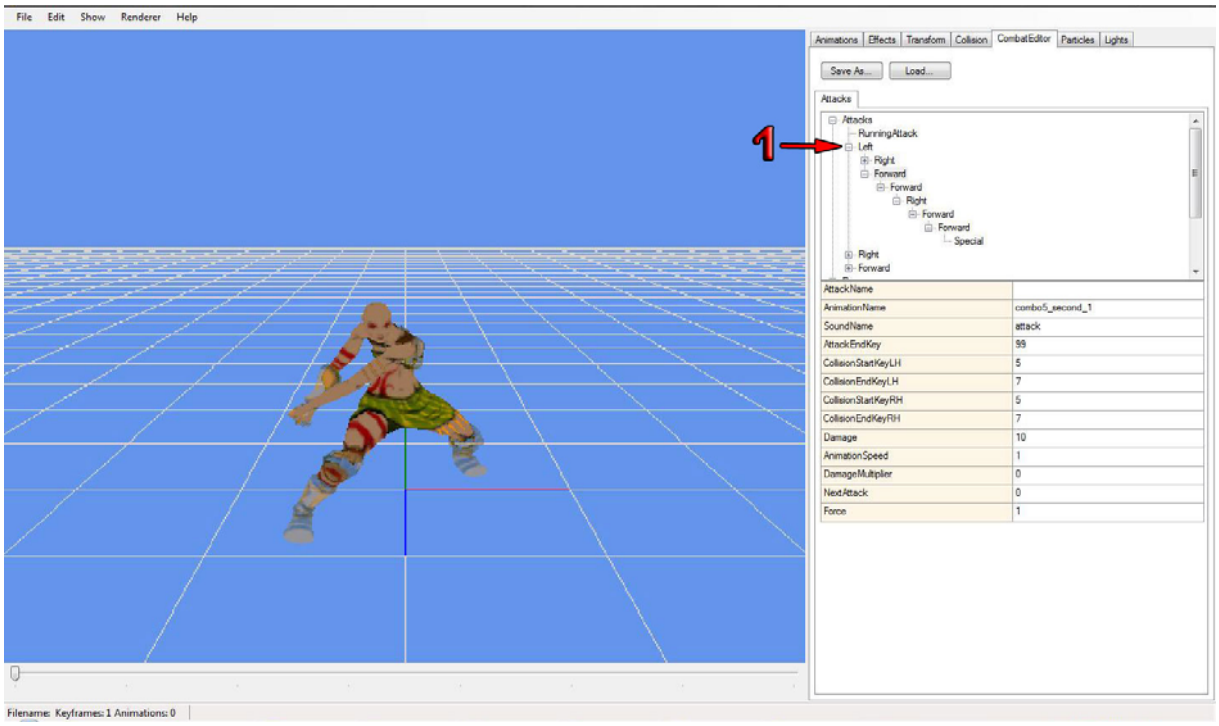


Figur 3: En bild på CombatEditor och en stils (Mace) trädstruktur (ej expanderad).

På bilden (*Figur 3*) ovan visas samtliga attacker för stilen *Mace*. Alla attacker i spelet följer en trädstruktur, d.v.s varje nästkommande attack i en combo utgår ifrån föregående. I praktiken innebär detta att spelaren inte kan utföra *Attack B* utan att **först** utföra *Attack A* (där *A* och *B* representerar de två första attackerna i en combo).

I bilden (*Figur 3*) syns även de olika sorternas attacker: *Basic attacks* (**1**), *Combos* (varje + representerar minst en combo – se nedan), *Rage attacks* (**2**) samt *Charge attacks* (**3**).

Nedanstående bild (*Figur 4*) illustrerar utseendet på trädstrukturen när en combo ”fällts ut”.



Figur 4: En bild på CombatEditor och den expanderade trädstrukturen hos en stil (Mace).

Det första som bör studeras i ovanstående bild (*Figur 4*) är hur trädstrukturen expanderats (användaren har klickat på ett av plussen). Bilden visar alltså att en *Basic attack* av typen l*eft* (**1**) kan leda till två möjliga combos:

1. *Left – Right – …*
2. *Left – Forward – …*

Studeras den combo som inleds med *Left*, *Forward* syns något intressant med en gång. I verktyget listas den fullständiga combon som:

Left – Forward – Forward – Right – Forward – Forward – Special

Den input som spelaren i själva verket måste göra för att utföra combon består dock av:

Left – Forward – Right – Forward

Detta är ett bra exempel på hur de verktyg som används i det praktiska arbetet kan tvinga designers att revidera sitt tankesätt helt och hållet.

Skillnaden mellan design och praktik i detta fall är att i verktyget syns varje steg av attacken, inte bara de steg som krävs i input för att combon skall spelas upp. Anledningen till att *Forward* förekommer ”dubbelt” vid två tillfällen är att denna animation ”splittats”.

Splits

Tidigare (avsnittet Collision – s. 9) har det nämnts att kollision endast kan ske en gång per vapen och animation. Detta kan ställa till med problem för vissa combos, eftersom animationer ibland innehåller flera kollisioner.

Anledningen till detta är att vi i designen av samtliga combos eftersträvat ”explosivitet” och ”oberäknelighet” och därför angivit för gruppens animatör antalet kollisioner som skall kunna ske i varje animation för varje combo. Dokument där denna design presenteras finns inkluderade som bilagor till uppsatsen (Bilaga 1 och 2).

Den ursprungliga tanken bakom designen var att input alltid skall motsvaras av en ”trovärdig” visuell representation (**1** input motsvarar **1** slag) men efter diskussion med gruppens animatör fick denna tanke revideras. Denne ansåg det minst lika viktigt (om inte viktigare) att ha estetiskt tilltalande animationer och inte bara ”funktionella” sådana. Detta ledde till en revidering av designen och att vissa animationer ”överdrevs”.

I praktiken innebär detta att vissa animationer har fler kollisioner än de som motsvaras av angiven input (**1** input kan exempelvis motsvara **3** slag). Detta leder i förlängningen till att de slutgiltiga skadevärdena hos animationer med flera splits blir lägre (jämfört med andra attacker), eftersom antalet kollisioner är större. Ett exempel på detta redovisas i nästa avsnitt.

Animationen ovan, där *Forward* förekommer två gånger, har alltså ”överdrivits” och innehåller därför fler kollisioner än de som motsvaras av angiven input (**1** *Forward* **input** resulterar i **2** *Forward* **slag**). För att båda dessa slag faktiskt skall kunna träffa en motståndare måste animationen *splittas*. Animationen måste alltså delas upp en gång för varje möjlig kollision (i detta fallet 2). Hade ovanstående attack inte splittats hade det i spelet sett ut som att två slag skett men endast en träff hade registrerats.

Själva processen att faktiskt dela upp en animation i flera mindre har utförts i en annan del av verktyget men har ingen relevans för vidare förståelse av balansarbetet.

En kort sammanfattning av splits betydelse:

- Syftet med splits är att tillåta fler kollisioner än 1 per vapen under en viss combo.
  - *Innebär i praktiken att en animation delas upp i flera, mindre animationer.*
  - *Tillåter att animationer ”överdrivs” – blir mer estetiskt tilltalande.*
  - *Splittade attacker = fler kollisioner = lägre slutgiltigt skadevärde.*

3.1.3 Parametrar

Nedan förklaras de parametrar i *CombatEditor* som haft stor betydelse för balansarbetet. De parametrar som ej förklaras anses inte vara av relevans för vidare förståelse och har därför exkluderats. I slutet av detta avsnitt förklaras den specifika betydelsen av varje parameter för balansarbetet.

AttackEndKey

I denna ruta anges det totala antalet effektiva key-frames. Siffran här motsvarar alltså det totala antal frames som måste spelas upp innan uppspelningen av nästa animation kan påbörjas. Denna siffra har stor betydelse för balans eftersom den behövs för att räkna ut tiden för varje combo.

I bilden ovan står siffran 99 som *AttackEndKey*. Detta eftersom animationen är den första ”splitten” i en kedja. Hela animationen måste således spelas upp (input kan inte ske förrän efter den sista splitten) och för att garantera detta sätts siffran till ett högt värde.

I praktiken innebär detta:

- *Antalet effektiva key-frames i första ”splitten” = **Samtliga** key-frames i animationen.*

CollisionStartKeyLH/RH – CollisionEndKeyLH/RH

Dessa parameterar ger spelet information om när kollision skall aktiveras respektive avaktiveras. *LH* betyder *Left Hand* och *RH* betyder *Right Hand*. Båda dessa parametrar behövs för *Dual-Wield*, eftersom denna stil använder två vapen simultant. Ifall kollision inte sker på en hand under en animation sätts parametrarna till 99/99, vilket i detta sammanhang avaktiverar kollision helt (kollisionen har i praktiken aktiverats och avaktiverats ”utanför” de key-frames som motsvarar själva animationen).

Damage

Här sätts skadevärdet per kollision för den befintliga animationen.

Sammanfattning – betydelse

Vad har då dessa parametrar för betydelse för själva balansarbetet?

AttackEndKey

Ger antalet effektiva key-frames vilket kan användas att ta reda på totala minimitiden för en combo. Summan av samtliga *AttackEndKey:s* i en combo är alltså det minsta antal key-frames som måste spelas upp innan combon avslutats. Genom att addera dessa siffror ges alltså den ”optimala” tiden för en combo, d.v.s hur lång tid en combo tas ifall tajmingen av dess input sker ”perfekt”.

Denna siffra är en av de variabler som används vid balansering av combos gentemot varandra, dess betydelse beskrivs i större detalj i ett kommande avsnitt (Teori - "Hastighet VS Skada" – s. 19).

CollisionStart/EndKeyLH/RH

Ger i förlängningen antalet kollisioner per ”steg” i en combo och påverkar således skadevärdet. Detta är viktigt att ta i beaktning eftersom desto fler kollisioner/splits, desto lägre skadevärde per animation/split.
Ett teoretiskt exempel:

Dual-Wield har en combo vars skadevärde vid en given animation skall vara 50 enheter. Animationen är tänkt att innehålla 4 kollisioner och måste därför splittas i 2 delar (kollision kan ske 1 gång per vapen och animation – *Dual-Wield* har 2 vapen). Varje split innehåller nu alltså 2 kollisioner.

Det totala antalet kollisioner blir därför 4 (2 x 2) och skadevärdet per split blir således: 50/4 = **12,5**.

Detta eftersom:

- Split 1
  - *Kollision 1 = 12,5 skada*
  - *Kollision 2 = 12,5 skada*
  - *12,5 + 12,5 = 25*
- Split 2
  - *Kollision 1 = 12,5 skada*
  - *Kollision 2 = 12,5 skada*
  - *12,5 + 12,5 = 25*
- **Split 1 + Split 2 = 25 + 25 = 50 skada**

Notera även att ovanstående exempel inte tagit hänsyn till om attackerna kan utföras som swings eller inte (se avsnitt Swings – s. 5). Om vi gör antagandet att attackerna faktiskt kan ”swingas” (de är av typen left eller right) måste nu skadevärdet halveras.

Det ”nya” skadevärdet per split skulle därför bli 12,5/2 = 6,25 enheter.

Det ter sig tämligen uppenbart att det finns ett flertal olika variabler att ta hänsyn till vid angivande av skadevärden. En sammanfattning av samtliga faktorer med ett praktiskt exempel återfinns därför i ett senare kapitel (Balansarbete – s. 18).

Damage
Här sätts själva skadevärdet för varje animation, vilket till stor del styrs av antalet kollisioner, som kan ses ovan.

Alltså:

- **AttackEndKey**
  - *Ger ”optimala” tiden för en animation och i förlängningen hela combon.*
  - *Påverkar den totala tiden för en combo, som i sin tur påverkar skadevärden.*
- **CollisionStart/EndKeyLH/RH**
  - *Ger antalet kollisioner.*
  - *Påverkar vilket skadevärde som skall anges för varje animation/attack.*
- **Damage**
  - *Bestämmer skadan per animation/attack.*
  - *Resultatet av balansarbetet.*

4 Balansarbete

I detta avsnitt sammanfattas och förklaras de processer och steg som behöver tas hänsyn till vid balansering av attacker mellan *Dual-Wield* och *Mace*. Avsnittet består till stor del av förklarande text men innehåller även ett praktiskt exempel, som används för att illustrera arbetet i sig.

Implementerade värden finns bifogade med arbetet (Bilaga 3) och tanken är att läsaren efter att ha läst detta avsnitt själv skall kunna förstå motiveringen bakom satta värden.

4.1 Generell tanke

Som nämnts tidigare (under avsnittet Project Colosseum – s. 2) var den ursprungliga tanken att ställa stilarna i kontrast till varandra på följande sätt:

- **Mace**
  - *Långsam.*
  - *Gör mycket skada.*
- **Dual-Wield**
  - *Snabb.*
  - *Gör lite skada.*

Det praktiska arbetet (bestämmandet av värden) har således gått ut på att förverkliga denna tanke.

4.2 Begränsningar

Rent teoretiskt hade det varit möjligt att ändra på animationslängd ifall det skulle krävas för att lättare kunna balansera attacker mot varandra (se till att den totala ”optimala” tiden liknar varandra så mycket som möjligt) men balansarbetet har haft sin grund i att detta inte är ett gott praktiskt alternativ.

Det är möjligt men absolut inte föredraget, på grund av den tidsbrist som råder i utvecklingen. Detta innebär att de värden som gäller för key-frames bör betraktas som ”absoluta”. Innebörden av detta påstående blir alltså att dessa värden är konstanter och därmed inte kan ändras på.

Balansen kan därför sägas bero på de ursprungliga värdena hos antalet effektiva key-frames för varje attack/combo.

4.3 Antaganden

Den spontana tanken när balanseringen påbörjades var att det finns ett ofantligt antal faktorer att ta hänsyn till förutom just skada och hastighet, alla med sin individuella påverkan på stilen. Exempel på sådana faktorer är spelarskicklighet samt den input som krävs för att utföra en attack. Dessa faktorer diskuteras i viss mån i ett kommande kapitel (Diskussion – s. 40) men för att avgränsa det praktiska arbetet har följande antaganden gjorts:

4.3.1 Skicklighet

Spelarskickligheten räknas inte som en avgörande faktor för arbetet. Arbetet baseras på att de spelare som möter varandra är ”lika skickliga”, vilket innebär att vissa faktorer ”försvinner ur ekvationen”. Ett exempel på en sådan faktor är input. Svårighetsgraden vid utförandet av en combo har således inte tagits med i beräkningen av skadevärden.

4.3.2 Optimal input

Trots att spelarskicklighet inte har tagits hänsyn till som en avgörande faktor för att specificera värden för varje individuell attack har det ändå viss vikt för balanseringen. Samtliga värden på effektiva key-frames är baserade på optimal input, d.v.s. att nästkommande input i en combo utförs med perfekt timing – direkt efter den sista frame:n.

Om en animation pågår 18 key-frames skulle perfekt timing alltså innebära att nästa input sker på key-frame 19, 1/25 sekund efter det att attacken slutförts (detta eftersom animationshastigheten är 25 FPS – se avsnitt Key-frames – s. 11). I praktiken stämmer naturligtvis inte detta påstående men för arbetet i sig har det haft stor betydelse.

Detta eftersom varje attack i båda stilar måste jämföras gentemot varandra på samma villkor. För enkelhetens skull har därför det optimala värdet använts som ”riktvärde”. Den praktiska innebörden av detta påstående blir att den totala tiden för en attack blir summan av dess effektiva key-frames.

4.4 Teori – ”Hastighet VS Skada”

Målet med arbetet är i grund och botten att se till att båda stilar hamnar i balans med varandra. I praktiken innebär detta att oavsett vilka stilar som möts så skall båda ha lika stor chans att vinna en match.

Om den grundläggande mekaniken i ett fighting-spel skall definieras handlar det i grund och botten om att åsamka sin motståndare mer skada än denne åsamkar dig. För att stilarna *Dual-Wield* och *Mace* skall kunna vara jämbördiga innebär detta alltså att följande påstående måste uppfyllas:

- *Samtliga attacker för båda stilar, jämförda mot dess motsvarighet för den andra stilen, skall kunna åsamka lika stor mängd skada.*

Ovanstående påstående stämmer väl förutsatt att samtliga attacker har samma värde på det totala antalet effektiva key-frames men inte annars. Två attacker av samma typ men med olika faktiska tidslängd kan inte ha samma skadevärde. Detta eftersom skadan måste sättas i relation till den faktiska tiden en animation tar att spela upp.

Ett exempel (på ett fiktivt extremfall):

- *Attack A – Dual-Wield*
  - Skadevärde = 60
  - Totalt antal effektiva key-frames = 30
  - Skada/frame = 60/30 = 2

- *Attack A – Mace*
  - Skadevärde = 60

- Totalt antal effektiva key-frames = 55
- Skada/frame = 60/55 = 1,1

Båda attacker har samma skadevärde men olika skada per key-frame. Inte nog med detta, Attack A för *Mace* är 25 key-frames längre, vilket innebär att den i spelet tar 1 sekund längre att spela upp jämfört med motsvarande attack för *Dual-Wield*. Detta innebär att *Dual-Wield* hinner utföra nästan 2 attacker på samma tid som det tar *Mace* att utföra 1!
Den potentiella skadan är alltså i teorin dubbelt så stor, vilket illustreras väl av skillnaden mellan skada/frame (2-1,1 = 0,9 = 90 % mer = nästan dubbelt så stor skada).

Ovanstående exempel illustrerar hur avgörande tiden är som faktor för det slutgiltiga bestämmandet av skadevärde.

Det har redan konstaterats att det totala antalet key-frames per attack är ett konstant värde och detta innebär alltså att balans av stilar enligt denna modell sker uteslutande genom ändringar i skadevärden.

Ovanstående exempel illustrerat med ”balanserade” skadevärden:

- *Attack A – Dual-Wield*
  - Skadevärde = **33**
  - Totalt antal effektiva key-frames = 30
  - Skada/frame = 33/30 = 1,1

- *Attack A – Mace*
  - Skadevärde = 60
  - Totalt antal effektiva key-frames = 55
  - Skada/frame = 60/55 = 1,1

En förklaring till hur beräkningen utfördes:

1. *Det minsta antalet effektiva frames (**30**) delades med det andra värdet (högsta antalet = **55**).*
2. *Resultatet blev ett decimaltal som motsvarar den procentuella relationen mellan antalet key-frames (30/55 = **0,55**).*
3. *Decimaltalet multiplicerades med skadevärdet hos attacken med högst antal effektiva frames (**0,55** x **60**).*
4. *Resultatet motsvarar det balanserade värdet (0,55 x 60 = **33**).*

Det viktiga att förstå i ovanstående uträkning är hur tiden påverkar skadevärdet.
En formel som illustrerar hur olika värden påverkar varandra ser ut på följande sätt:

***Minsta** antal effektiva frames / **högsta** antalet effektiva frames.*

Resultatet blir ett decimaltal som motsvarar skillnaden i hastighet mellan attackerna – faktorn mellan attackerna. I exemplet ovan visar resultatet att *Attack A* för *Dual-Wield* bara tar 0,55 = 55 % av den tiden det tar för motsvarande attack att utföras för *Mace*.

Genom att multiplicera detta värde med skadevärdet för *Mace* balanseras skadan hos *Dual-Wield*. Det ursprungliga skadevärdet minskas med 45 % (1 – 0,55) och hamnar därför i balans.

Med andra ord har det balanserade skadevärdet sitt ursprung i den tid det tar för animationen att spelas upp.

Påståendet för balans i denna situation blir således:

A x B, där:

A = Faktor (se ovan).
B = Ursprungligt skadevärde.

För att denna uträkning skall fungera måste således ett ursprungligt skadevärde finnas. Detta har nämnts tidigare (se kapitel Attacktyper – s. 4) men kommer nu att konkretiseras ännu mer.

4.4.1 Riktvärden

I ovanstående exempel jämfördes antalet key-frames för *Dual-Wields* attack med motsvarande för *Mace* och resultatet från detta jämfördes sedan med skadevärdet hos *Mace*. De slutgiltiga värdena för *Dual-Wield* berodde med andra ord på redan befintliga värden hos *Mace* – de så kallade riktvärdena.

Eftersom riktvärden behövs för varje attack har *Mace* använts som ett ”generellt riktvärde”. Detta innebär i praktiken att skadevärden bestäms för *Mace* från början och därefter anpassas skadevärden för *Dual-Wield* utefter detta.

Om en attack för *Mace* har ett skadevärde på 50 enheter fungerar detta alltså som riktvärde – skadevärdet för *Dual-Wield* har sin utgångspunkt i detta värde.

Motiveringen till att *Mace* valts som generellt riktvärde grundar sig i den ursprungliga tanken om stilarnas ”beteende”, där *Mace* skulle vara betydligt långsammare än *Dual-Wield* men samtidigt göra mer skada.

Notera att det är möjligt att balansera ”åt motsatt håll” ifall det skulle behövas. Ifall en attack/combo för *Dual-Wield* har fler antal effektiva key-frames är det alltså möjligt att öka dess skadevärde för att uppnå balans. Balansering kan med andra ord ske ”åt båda hållen” om nödvändigt. Tanken med att använda *Mace* som generellt riktvärde är antagandet att det i majoriteten av fallen kommer krävas en sänkning av skadevärdet hos *Dual-Wield* (stilen är snabbare).

4.4.2 Interna riktvärden

Noterbart är att riktvärden inte enbart gäller stilarna emellan utan även har en viktig funktion vid balansering av interna skadevärden. En stil kan ha flera attacker/combos av samma typ och dessa måste även göra samma totala mängd skada.

Ett praktiskt exempel med utgångspunkt i *Basic Attacks* för *Mace*:

Varje *Basic Attack* för *Mace* skall ha ett skadevärde (i enheter) som motsvarar 2,5 % av totalen.

Mace Basic attacks = 0,025 x 700 = **18** enheter.

Efter denna generella uträkning måste samtliga *Basic attack*s jämföras internt, d.v.s. gentemot varandra:

Left-attack = **12** effektiva key-frames
Right-attack = **12** effektiva key-frames
Forward-attack = **13** effektiva key-frames.

I detta fall är alltså *Forward-attack* långsammast av attackerna. Denna attack fungerar därför som ett **internt** riktvärde (18 enheter), som sedan *Left-* och *Right-attack* balanseras utifrån.

Eftersom *Left-* och *Right-attack* har samma antal effektiva key-frames får de samma skadevärde.

Precis som i föregående avsnitt måste nu alltså *faktorn* för attacktyperna räknas ut. Denna blir då:

***Minsta** antal effektiva frames / **högsta** antalet effektiva frames.* →

→ 12/13 = **0,9** = 90 % av riktvärdet.

Faktorn multipliceras sedan med det ursprungliga skadevärdet/riktvärdet. →

→ 0,9 x 18 = **16** → Skadevärde för Left- & Right-attacks.

Notera även att *Left-* och *Right-attacks* kan utföras som swings (se avsnitt Swings – s. 5) vilket innebär att skadevärdet potentiellt kan dubblas. Det nya, framräknade värdet måste alltså halveras, för att balansen skall upprätthållas.

Det slutgiltiga skadevärdet för dessa attacker blir därför:

16/2 = **8**

Left-attack = **8** enheter.
Right-attack = **8** enheter.
Forward-attack = **18** enheter.

Att jämföra attacker såväl internt som externt är väldigt viktigt, eftersom det i förlängningen kan innebära att combos/attacker av samma typ men med olika antal effektiva key-frames får olika skadevärden, som illustreras av exemplet ovan. Det ursprungliga skadevärdet för *Left-* och *Right-attacks* har alltså minskat (från 9 till 8) efter jämförelse gentemot övriga attacker av samma typ, detta p.g.a. deras olika antal effektiva key-frames.

4.4.3 Ökning av skadevärden i Mace-combos

I avsnittet Combotyper (s. 6) beskrivs den generella designmässiga tanken bakom skadeökningen för de olika combotyperna. Att arbeta fram en formel för skadeökning för varje slag i varje combotyp skulle kräva mycket tid och resurser. Försök har dessutom gjorts och resultaten har inte alltid blivit så bra.

Det är med andra ord svårt att motivera hur eller varför de värdena som finns har bestämts, utan läsaren får helt enkelt acceptera att skadevärden för varje steg i *Mace*-combos är godtyckliga.

Den enda riktlinje som funnits vid bestämmandet av dessa värden har således varit att det sammanlagda skadevärdet ej får överstiga den förbestämda procentsatsen (exempelvis *Light Combos* = **15 %** av totalen – se kapitel Attacktyper – s. 4).

4.5 Tabeller

Nedan presenteras ett antal olika tabeller, med syfte att praktiskt redovisa det arbete som förklarats i tidigare avsnitt.

4.5.1 Effektiva key-frames

Nedanstående tabeller (Tabell 2 och 3) redovisar det totala antalet effektiva key-frames för samtliga attacktyper, både för *Mace* och för *Dual-Wield*.

Tabell 2 – antal effektiva key-frames för de olika attacktyperna (*Mace*)

Attacktyp	Totalt antal effektiva key-frames
Left attack	**12**
Right attack	**12**
Forward attack	**13**
Combo 1 (Light)	**56**
Combo 2 (Heavy)	**58**
Combo 3 Heavy)	**67**
Combo 4 (Advanced – ”vanlig” skada)	**65**
Combo 4 (Advanced – Special attack)	**51**
Combo 5 (Advanced – ”vanlig” skada)	**84**
Combo 5 (Advanced – Special attack)	**50**
Combo 6 (AOE)	**51**
Rage attack	**50**
Charge attack (Left)	**40**
Charge attack (Right)	**41**
Charge attack (Forward)	**19**

Tabell 3 – antal effektiva key-frames för de olika attacktyperna (*Dual-Wield*)

Attacktyp	Totalt antal effektiva key-frames
Left attack	**8**
Right attack	**8**
Forward attack	**10**
Combo 1 (Light)	**43**
Combo 2 (Light)	**38**
Combo 3 Heavy)	**54**
Combo 4 (Advanced – ”vanlig” skada)	**59**
Combo 4 (Advanced – Special attack)	**34**
Combo 5 (Advanced – ”vanlig” skada)	**72**
Combo 5 (Advanced – Special attack)	**52**
Combo 6 (AOE)	**56**
Rage attack	**39**
Charge attack (Left)	**22**
Charge attack (Right)	**23**
Charge attack (Forward)	**21**

Notera att vid jämförelse av antal effektiva key-frames gäller följande:

- Combos jämförs med den motsvarighet från motsatt stil där skillnaden i antal key-frames är **minst**.
    - *Exempel:*
        - *Combo 4 (Mace) – Combo 4 (Dual-Wield).*
- Finns fler combos av samma typ för en stil balanseras motsvarande attack för motsatt stil utifrån snittvärdet av totala antalet effektiva key-frames för alla combos av den typen.
    - *Exempel:*
        - *Combo 1 (Mace - Light) = **56** key-frames.*
        - *Jämförs mot (Combo 1 + Combo 2) / 2 (Dual-Wield - Light) →*
            - *(43 + 38) / 2 = **41** key-frames.*

4.5.2 Uppdaterade skadevärden – intern balansering

Nedanstående tabeller (Tabell 4 och 5) redovisar relationerna mellan attacker av samma typ men med olika antal effektiva key-frames, både för *Mace* och för *Dual-Wield*. I denna tabell redovisas även det uppdaterade totala skadevärdet för varje combo, jämfört med sin *interna* motsvarighet.

Värden redovisas ej för de attacker som fungerat som riktvärden (högst antal effektiva key-frames), eftersom dessa behåller sina ursprungliga värden. Notera även att det här inte tas hänsyn till swings vid uträkning av skadevärde på *Basic attacks*, detta görs vid ett senare tillfälle.

Tabell 4 – uppdaterade skadevärden i förhållande till antal effektiva key-frames (*Mace*)

Attacktyp	Totalt antal effektiva key-frames	Relation (jämfört med riktvärdet)	Uppdaterat skadevärde
Left attack	**12**	12 / 13 = **0,9**	0,9 x 18 = **16**
Right attack	**12**	12 / 13 = **0,9**	0,9 x 9 = **16**
*Forward attack (**riktvärde**)*	**13**	**-**	**-**
Combo 2 (Heavy)	**58**	58 / 67 = **0,87**	0,87 x 175 = **152**
*Combo 3 (Heavy - **riktvärde**)*	**67**	**-**	**-**
Combo 4 (Advanced - ”vanliga” attacker)	**65**	65 / 84 = **0,77**	0,77 x 70 = **54**
*Combo 5 (Advanced - ”vanliga” attacker - **riktvärde**)*	**84**	**-**	**-**
*Combo 4 (Advanced – Special attack – **riktvärde**)*	**51**	**-**	**-**
Combo 5 (Advanced – Special attack)	**50**	50 / 51 = **0,98**	0,98 x 210 = **206**
Charge attack (Left)	**40**	40 / 41 = **0,98**	0,98 x 53 = **52**
*Charge attack (Right - **riktvärde**)*	**41**	**-**	**-**
Charge attack (Forward)	**19**	19 / 41 = **0,46**	0,46 x 53 = **24**

Tabell 5 – uppdaterade skadevärden i förhållande till antal effektiva key-frames (*Dual-Wield*)

Attacktyp	Totalt antal effektiva key-frames	Relation (jämfört med riktvärdet)	Uppdaterat skadevärde
Left attack	**8**	8 / 10 = **0,8**	0,8 x 18 = **14**
Right attack	**8**	8 / 10 = **0,8**	0,8 x 18 = **14**
*Forward attack (**riktvärde**)*	**10**	-	-
*Combo 1 (Light - **riktvärde**)*	**43**	-	-
Combo 2 (Light)	**38**	38 / 43 = **0,88**	0,88 x 105 = **92**
Combo 4 (Advanced - ”vanliga” attacker)	**59**	59 / 72 = **0,82**	0,82 x 70 = **57**
*Combo 5 (Advanced - ”vanliga” attacker - **riktvärde**)*	**72**	-	-
Combo 4 (Advanced – Special attack)	**34**	34 / 52 = **0,65**	0,65 x 210 = **137**
*Combo 5 (Advanced – Special attack – **riktvärde**)*	**52**	-	-
Charge attack (Left)	**22**	22 / 23 = **0,96**	0,96 x 53 = **51**
*Charge attack (Right - **riktvärde**)*	**23**	-	-
Charge attack (Forward)	**21**	21 / 23 = **0,9**	0,9 x 53 = **48**

Notera att samtliga skadevärden för *Dual-Wield* inte ännu är balanserade. De värden som använts är alltså de giltiga för *Mace* och ovanstående värden kommer senare att balanseras om utifrån dessa. Tabellen ovan syftar snarare på att visa principen bakom balanseringen.

4.5.2 Balansering – Basic attacks (Dual-Wield)

Nedanstående tabell (Tabell 6) redovisar balanserade skadevärden för samtliga Basic attacks för Dual-Wield. Dessa data presenteras redan i detta kapitel eftersom de är av största vikt för angivandet av skadeökningen i samtliga combos.

Tabell 6 – balanserade skadevärden för *Basic attacks* (*Dual-Wield*)

Attacktyp	Totalt antal effektiva key-frames	Relation (jämfört med riktvärdet)	Uppdaterat skadevärde
*Left attack (Mace – **riktvärde**)*	**12**	-	-
Left attack (Dual-Wield)	**8**	8 / 12 = **0,67**	0,67 x 16 = **11**
*Right attack (Mace – **riktvärde**)*	**12**	-	-
Right attack (Dual-Wield)	**8**	8 / 12 = **0,67**	0,67 x 16 = **11**
*Forward attack (Mace – **riktvärde**)*	**13**	-	-
Forward attack (Dual-Wield)	**10**	10 / 13 = **0,77**	0,77 x 18 = **14**

4.5.3 Skadeökning

Nedanstående tabeller (Tabell 7 och 8) redovisar den godtyckliga skadeökningen för varje slag i samtliga combos, både för Mace och för Dual-Wield. Noterbart är att den inledande attacken i varje combo är av typen Basic attack och därför alltid måste ha det värde som motsvarar dess typ (Left, Right eller Forward), oavsett vilken sorts combo det är.

Tabell 7 – skadeökning i stegen för varje combo (*Mace*)

Comboty p	Tota lska da	Skada – Steg 1	Skada – Steg 2	Skada – Steg 3	Skada – Steg 4
Combo 1	**105**	**18**	**37**	**50**	-
Combo 2	**152**	**16**	**21**	**40**	**75**
Combo 3	**175**	**18**	**30**	**52**	**75**
Combo 4 (”vanlig” skada)	**54**	**18**	**11**	**12**	**13**
Combo 5 (”vanlig” skada)	**70**	**16**	**17**	**18**	**19**
Combo 6	**88**	**16**	**27**	**45**	-

Tabell 8 – skadeökning i stegen för varje combo (*Dual-Wield*)

Combotyp	Totalskada	Skada – Steg 1	Skada – Steg 2	Skada – Steg 3	Skada – Steg 4	Skada – Steg 5	Skada – Steg 6
Combo 1	**105**	**14**	**36**	**55**	**-**	**-**	**-**
Combo 2	**92**	**14**	**30**	**48**	**-**	**-**	**-**
Combo 3	**175**	**11**	**34**	**55**	**75**	**-**	**-**
Combo 4 (”vanlig” skada)	**57**	**14**	**13**	**14**	**16**	**-**	**-**
Combo 5 (”vanlig” skada)	**70**	**11**	**10**	**11**	**12**	**12**	**14**
Combo 6	**88**	**11**	**29**	**48**	**-**	**-**	**-**

Notera att de uppdaterade skadevärdena för *Basic attacks* förts in i ovanstående tabell samt att ovanstående värden för *Dual-Wield* inte är att betrakta som slutgiltiga, eftersom de måste balanseras om gentemot *Mace*. De har dock ett stort värde, eftersom grundläggande värden för skadeökning fortfarande behövs.

4.6 Praktik – Hur räknar man?

Följande avsnitt syftar till att förklara för läsaren alla de steg som måste tas vid det praktiska arbetet med balansering. Vidare presenteras även ett praktiskt exempel för att illustrera arbetet i sig.

4.6.1 Balanseringsprocessen – en lathund

Följande steg är nödvändiga för att balansera en attack/combo gentemot dess interna eller externa motsvarighet:

Del 1

1. *Bestäm riktvärde – Vilken attack/combo skall jag utgå ifrån?*
2. *Räkna samman antal effektiva key-frames för båda attacker/combos.*
3. *Dela det lägsta resultatet med det högsta.*
4. *Multiplicera resultatet av divisionen med riktvärdet (totala skadevärdet för attacken/combon du utgår ifrån).*
    a. *Resultatet är det nya, balanserade **totala** skadevärdet.*

Om balansering sker mellan combos behövs ett antal steg till för att faktiskt kunna föra in de nya skadevärdena:

Del 2

5. *Subtrahera det nya, totala skadevärdet från riktvärdet.*
    a. *Differensen blir antal enheter som combon måste ökas/minskas med.*
6. *Räkna ut antalet ”steg” (animationer) i combon (som skall balanseras om).*
7. *Dela **differensen** med **antalet modifierbara steg (ej första – absolut värde)**.*
    a. *Resultatet är ett snittvärde av skadeenheter som **varje steg** i combon bör ökas/sänkas med för att balansera gentemot riktvärdet.*
8. *Addera/subtrahera snittvärdena på varje steg i den **nya** combon.*
    a. ***Kom ihåg:***
        - *Kontrollera antalet kollisioner för varje steg, dela skadevärdet med detta!*
            - ***2 Kollisioner = Skadevärde halveras.***
            - ***Splits kan ge flera kollisioner, ex. 3 st. = dela med 3!***
        - *Kontrollera slagtyp!*
            - ***Left/Right = Skadevärde halveras.***
        - *Avrunda ojämna tal uppåt.*
        - *Om Special-attack:*
            - *Räkna ut balanserat värde på samma sätt som i Del 1.*

4.6.2 Balanseringsprocessen – ett praktiskt exempel

För att läsaren i förlängningen skall kunna förstå arbetet bakom bestämmande av skadevärden illustreras nedan ett praktiskt exempel, där två combos av samma typ balanseras mellan stilarna. Samtliga processer i lathunden kommer att användas och vävs samman med texten för att tydliggöra hur själva processen går till.

1. *Bestäm riktvärde – Vilken attack/combo skall jag utgå ifrån?*

I detta exempel balanseras två *Advanced combos* gentemot varandra. Notera att balansering endast kommer ske på den ”vanliga” skadan, hänsyn tas alltså inte till combons *Special attack* i nedanstående exempel.

Riktvärdet i detta fall blir *Mace – Combo 5 (Left – Forward – Right – Forward).*

Skadevärden för denna combo (se tabeller ovan):

16 + 17 + 18 + 19

Combon som skall jämföras mot denne blir dess motsvarighet för *Dual-Wield – Combo 5 (Left – Forward – Right – Left – Right – Forward).*

Det märks redan nu att just balanseringen av dessa combos representerar ett intressant fall, eftersom de har olika antal inputs/”steg”.

2. *Räkna samman antal effektiva key-frames för båda attacker/combos.*

Antal effektiva key-frames för *Mace* (parenteser representerar splits):

- 12 + (8+20) + 16 + (8+20) = **84**

Antal effektiva key-frames för *Dual-Wield* (parenteser representerar splits):

- 8 + 9 + (3+9) + 15 + 17 + 11 = **72**

3. *Dela det lägsta resultatet med det högsta.*

72 / 84 = **0,86**

4. *Multiplicera resultatet av divisionen med riktvärdet (totala skadevärdet för attacken/combon du utgår ifrån).*

0,86 x 70 = **60**

5. *Subtrahera det nya, totala skadevärdet från riktvärdet.*

70 – 60 = **10**

6. *Räkna ut antalet ”steg” (animationer) i combon (som skall balanseras om).*

8 + 9 + (3+9) + 15 + 17 + 11 = **6** steg.

Notera att det inte tas hänsyn till splits i denna process, detta sker senare i beräkningen.

7. *Dela **differensen** med **antalet modifierbara steg (ej första – absolut värde)**.*

10 / 5 = **2**

8. *Addera/subtrahera snittvärdena på varje steg i den **nya** combon.*

Skadevärden för denna combo (*Dual-Wield – Combo 5*, se tabeller ovan):

11 + 10 + 11 + 12 + 12 + 14

Nu skall alltså snittvärdet (2) subtraheras från varje ”steg” (undantaget det första). Subtraktion sker istället för addition eftersom combon för *Dual-Wield* har **mindre** antal effektiva key-frames – attacken är **snabbare**.

Resultatet blir:

11 + (10-2) + (11-2) + (12-2) + (12-2) + (14-2) = **11 + 8 + 9 + 10 + 10 + 12**

Kontrollräkning visar att summan blir 60 → Uträkning stämmer!

Vi fortsätter vidare i listan:

- *Kontrollera antalet kollisioner för varje steg, dela skadevärdet med detta!*
  - ***2 Kollisioner = Skadevärde halveras.***
  - ***Splits kan ge flera kollisioner, ex. 3 st. = dela med 3!***

Antalet kollisioner för varje steg i en combo får av läsaren betraktas som någorlunda godtyckligt. Dessa antal har fåtts genom att studera varje animations utseende i *Toolkit*, en process som inte kommer att beskrivas närmre. Läsaren får därför helt enkelt göra antagandet att det antal kollisioner som anges stämmer!

Vi börjar med *Mace*:

- *Left – Forward – Right – Forward*
- Antal kollisioner per steg:
  - *Steg 1 (Left) = 1 kollision*
  - *Steg 2 (Forward) = **2** kollisioner*
  - *Steg 3 (Right) = 1 kollision*
  - *Steg 4 (Forward) = **2** kollisioner*

De värden som finns i Steg 2 & 4 måste alltså halveras (delas med 2):

- Steg 2 (17 enheter) = 17/2 = **8,5**
- Steg 4 (19 enheter) = 19/2 = **9,5**

Reviderade skadevärden *(Mace – Combo 5)***:**

16 + **8,5** + 18 + **9,5**

Notera här att avrundning av tal inte sker förrän det säkerställts att inga fler förändringar av värdet kommer att ske.

Samma procedur upprepas för *Dual-Wield*:

- *Left – Forward – Right – Left – Right – Forward*
- Antal kollisioner per steg:
  - *Steg 1 (Left) = 1 kollision*
  - *Steg 2 (Forward) = **2** kollisioner*
  - *Steg 3 (Right) = **2** kollisioner*
  - *Steg 4 (Left) = 1 kollision*
  - *Steg 5 (Right) = **2** kollisioner*
  - *Steg 6 (Forward) = 1 kollision*

De värden som finns i Steg 2, 3 & 5 måste alltså halveras (delas med 2):

- Steg 2 (8 enheter) = 8/2 = **4**
- Steg 3 (9 enheter) = 9/2 = **4,5**
- Steg 5 (10 enheter) = 10/2 = **5**

Reviderade skadevärden *(Dual-Wield – Combo 5)***:**

11 + **4** + **4,5** + 10 + **5** + 11

Vi fortsätter vidare:

- *Kontrollera slagtyp!*
  - ***Left/Right = Skadevärde halveras.***

Detta steg måste alltså utföras för **båda** stilar.

*Mace – Combo 5***:**

- ***Left** – Forward – **Right** – Forward*
- Steg 1 & 3 halveras!
  - *Steg 1 (16 enheter) = 16/2 = 8*
  - *Steg 3 (18 enheter) = 18/2 = 9*
- Reviderade skadevärden:
  - **8** + 8,5 + **9** + 9,5

*Dual-Wield – Combo 5***:**

- ***Left** – Forward – **Right** – **Left** – **Right** – Forward*
- Steg 1 (11 enheter), 3 (4,5 enheter), 4 (10 enheter) & 5 (5 enheter) halveras!
  - *Steg 1 (11 enheter) = 11/2 = 5,5*
  - *Steg 3 (4,5 enheter) = 4,5/2 = 2,25*
  - *Steg 4 (10 enheter) = 10/2 = 5*
  - *Steg 5 (5 enheter) = 5/2 = 2,5*
- Reviderade skadevärden:
  - **5,5** + 4,5 + **2,25** + **5** + **2,5** + 11

Sist men inte minst:

- *Avrunda ojämna tal uppåt.*

Uträkningen, tillsammans med det slutgiltiga resultatet presenteras i tabellerna nedan (Tabell 9 och 10). Resultatet för varje ”steg” representerar alltså det skadevärde som skall föras in på samtliga *Damage*-poster i *Toolkit* för varje ”steg” i en combo. Om ett steg består av flera splits skall alltså samma värde föras in på samtliga poster (i det steget)!

Notera avslutningsvis att avrundning sker i den sista rutan (*Slutgiltigt värde*), varför vissa resultat ej innehåller de decimaler de ”borde”.

Tabell 9 – slutgiltiga skadevärden för varje steg (*Mace – Combo 5*)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**16**	**1**	16 / 1 = **16**	**J**	16 / 2 = **8**
Steg 2 (Forward)	**17**	**2**	17 / 2 = **8,5**	**N**	9 / 1 = **9**
Steg 3 (Right)	**18**	**1**	18 / 1 = **18**	**J**	18 / 2 = **9**
Steg 4 (Forward)	**19**	**2**	19 / 2 = **9,5**	**N**	9,5 / 1 = **10**

Tabell 10 – slutgiltiga skadevärden för varje steg (*Dual-Wield – Combo 5*)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**11**	**1**	11 / 1 = **11**	**J**	11 / 2 = **6**
Steg 2 (Forward)	**8**	**2**	8 / 2 = **4**	**N**	4 / 1 = **4**
Steg 3 (Right)	**9**	**2**	9 / 2 = **4,5**	**J**	4,5 / 2 = **2**
Steg 4 (Left)	**10**	**1**	10 / 1 = **10**	**J**	10 / 2 = **5**
Steg 5 (Right)	**10**	**2**	10 / 2 = **5**	**J**	5 / 2 = **3**
Steg 6 (Forward)	**12**	**1**	12 / 1 = **12**	**N**	12 / 1 = **12**

Det allra sista steget i lathunden, att balansera skadevärdet hos *Special-attacks* kommer inte att illustreras på samma sätt, eftersom det principiellt fungerar på samma sätt som att balansera en attack utefter Del 1 (s. 29, vilket redan illustrerats).

Dessa värden och samtliga uträkningar finns inkluderade som en bilaga till uppsatsen. Läsaren bör dock vara medveten om att uträkningar för hur det balanserade värdet fås fram inte finns representerat i bilagan. För full förståelse krävs alltså att denne följer samtliga stegen i lathunden! Det framgår med all tydlighet att denna process är både komplicerad och tidskrävande. De fördelar och nackdelar förknippade med detta arbete diskuteras i uppsatsens avslutande kapitel.

5 Teorier om balansarbete

I detta kapitel presenteras ett antal tankar och teorier rörande spelbalans som känns relevanta satta i relation till det utförda arbetet. En diskussion kring de olika teorierna återfinns i uppsatsens avslutande kapitel.

5.1 Patterns for Game Mastery and Balancing

I sin bok *Patterns in Game Design (Björk & Holopainen, 2005)* skriver författarna om vikten av balans i spel.

”*For games where players play against opponents, the players need to feel that they can affect the outcome of the game. If a game is designed with a certain game time or amount of gameplay, and players feel powerless, these players have two possibilities: endure gameplay that is uninspiring or suffer gameplay breakdown due to the players' desire to stop playing.*”

(Björk & Holopainen, 2005, s. 387)

Författarna menar alltså att balans i spel är av yttersta vikt för spelarens underhållning. Ett obalanserat spel leder i förlängningen till att spelaren ”härdar ut” (spelar trots att de inte finner det underhållande) eller känner en stark vilja att sluta spela.

Vidare diskuteras två olika typer av balansering: *Preemptive* och *correcting*.

Distinktionen mellan dessa beskrivs på följande sätt:

”*Preemptive balancing effects try to maintain player balance so that imbalances do not occur, while correcting balancing effects try to correct imbalances when they have occured.*”

(Björk & Holopainen, 2005, s. 387)

Vidare diskuterar författarna konsekvenserna av balansering:

”*The presence of balancing effects strengthens or prolongs players' perceived chance to succeed but lessens the perceivable margins of the game and removes feelings of game mastery in the game. “*

(Björk & Holopainen, 2005, s. 389)

Författarna menar alltså att balansering kan ske antingen i förebyggande syfte eller efter hand, i syfte att motverka uppkomna skiljaktigheter i spelets balans. De menar även att balans i ett spel i förlängningen gör att spelaren får större tilltro till den egna förmågan men att bemästringsfaktorn minskar. Spelarna tror sig alltså klara spelet, men inte ”bra”.

5.2 The Internal Economy of Games and Game Balancing

I boken O*n Game Design* (Rollings & Adams, 2003) beskriver författarna sin syn på vad ett balanserat spel är:

”*But what exactly do we mean by a balanced game? A balanced game is one where the main determining factor for the success of the player is the skill level of that player. That does not mean that random events cannot occur, but a better player should ordinarily be more successful than a poor one unless he has an unusually long run of bad luck.*”

(Rollings & Adams, 2003,s. 240)

Författarna anser alltså att spelarna i ett balanserat spel måste förlita sig på sin egen skicklighet som den faktor som avgör ifall de kommer att vinna eller förlora. Slumpmässiga händelser kan fortfarande ske men de skall inte vara avgörande för spelarens framgång, förutsatt att en spelare inte har ”extrem otur”.

Vidare diskuteras balansarbete och dess svårigheter generellt:

”*So why are there no formalized, scientifically rigorous methods of game balancing? Well, for a start, it’s an extremely difficult and complex process. Essentially, game balancing is a problem involving a fantastically large number of independent variables. It’s an optimization problem in n-dimensional space, where n is a very large number. No formal rules govern game balancing, except in a very small number of abstract mathematical scenarios.*”

“*This sort of n-dimensional optimization problem occurs in many areas of science, and it is from these areas that we can borrow techniques to help us solve the problem. This is not to say that all game-balancing problems can be solved by the blanket application of a sterile algorithm. A healthy measure of human finesse is still required in order to make a game feel “just right”. Just because a result is mathematically correct does not automatically make it aesthetically pleasing.*”

(Rollings & Adams, 2003, s. 241)

Enligt författarna är balansering av spel extremt svårt eftersom det antalet variabler som kan påverka varandra i teorin kan vara oändligt stort. Det finns inga ”generella magiska formler” för balansering och även om dessa fanns bör man som utvecklare inte förlita sig helt på dessa. Rollings & Adams (2005) menar att mänsklig finess också är en avgörande faktor och lyfter fram estetikens betydelse för spelaren. Balansering av spel handlar således både om matematisk balans och ”fingertoppskänsla” enligt dessa två herrar.

Statisk och *dynamisk* balans förklaras även:

”*Static balance is concerned with the rules of the game and how they interact with each other. These are specifically time invariant. In other words, these are the rules of the game that can be written down and documented to aid play – the usual strategy guide fodder. /…/Generally, when game balance is spoken about, most people are unconsciously referring to the static balance (sometimes mixed with a little of the dynamic balance).*”

(Rollings & Adams, 2003, s. 243)

"*Dynamic balance covers the opening, midgame and endgame of classic game analysis on a much finer scale. Rather than treating the game as three discrete phases, which is fine for postgame analyses, we have to consider the fully continuous spectrum of play, from start to end. This differs from the static balance, because we have to consider the interaction of the players or players with the statically balanced system. We are concerned not only with static balance, but also dynamic balance – how the balance changes with time and player interaction.*"

(Rollings & Adams, 2003, s. 267)

Författarna menar alltså att det finns två sorters balans att ta hänsyn till, den statiska och den dynamiska. Statisk balans handlar i grund och botten om specifika regler i spelet, som alla spelare måste rätta sig efter medan dynamisk balans behandlar spelarnas interaktion med spelet och den statiska balansen.

Vidare förklaras en (enligt författarna) viktig princip vid utveckling för att förenkla balansering – att designa för modifikation:

"*When designing a game, it is sensible to design a core set of rules that the game adheres to and then design the game entities to conform to those rules. This is often a simpler and more intuitive approach than designing entities that each requires its own special considerations. Not only does this make matters simpler when it comes to programming the game – the developers can concentrate on implementing the core rules and then adding entities on top of those core rules, rather than coding each entity separately – it also simplifies tweaking the design.*"

(Rollings & Adams, 2003, s. 281)

Rollings & Adams menar alltså här att en förutsättning för god balans är att spelets har en stabil designmässig grund att stå på. Samtliga "entiteter" i spelet bör rätta sig efter samma "kärnregler", både för att förenkla programmering och vidare tweaking [8]av dessa.

Avslutningsvis diskuterar författarna hur de anser att manipulering av värden bör ske:

"*One of the most important rules to bear in mind is that tweaking parameters randomly is an inefficient and wasteful way to modify balance. A brief digression into correct experimental method is required to ensure that you are making the best of your time.*"

"*The first, and most important, point to remember is to modify only one parameter at a time. Although it may be extremely tempting to tweak a whole bunch of parameters in order to force a result, unless you are extremely lucky, you won't get anything useful. /…/ When initially modifying parameters, don't bother with small changes; Brian Reynolds (of Big Huge Games) suggests doubling or halving the parameter and seeing what it does. From there, you can iteratively move toward the ideal value as efficiently as possible. This makes sense. By changing by such a large factor, you can easily zero in on your optimum setting.*"

(Rollings & Adams, 2003, s. 283)

[8] Att tweaka innebär i detta sammanhang att ändra på olika variabler, exempelvis skadevärden för attacker.

Författarna menar alltså att det praktiska balansarbetet (”tweakingen”) bör ske med eftertanke, för att (i den mån det är möjligt) försäkra sig om att man inte ”slösar tid i onödan”. De menar även att manipulering av parametrar bör ske individuellt och att när manipulering sker bör man initialt göra stora förändringar i värden, för att lättare kunna se effekterna av sina handlingar.

5.3 Player/Player Balance

I boken *Game Architecture and Design* (Rollings & Morris, 2004) diskuterar författarna (återigen) skicklighetens vikt i ett balanserat spel:

”*In most games, the designers aim is to arrange things so that victory is won by skill and judgement. We don’t mean that there mustn’t be an element of luck. Most strategies entail a gamble, and judging when a risk is worth taking is part of the fun. However, a balanced design should avoid random elements that favor one player, or whose effects are so magnified as the game progresses that victory comes mainly as a result of luck. The most glaring instance of such imbalance is when a player gets a lucky advantage right from the start – perhaps by starting with his city close to a gold mine, or by entering the game right outside the door of the laser arsenal. How is this kind of undue advantage best avoided?*”

(Rollings & Morris, 2004, s. 106)

Författarna menar alltså att slump i spel är godtagbart, så länge dessa är förknippade med en viss chanstagning. Dock bör ett balanserat spel aldrig tillåta att en spelare segrar som ett huvudsakligt resultat av tur. För att motverka dessa problem erbjuder författarna en lösning: *symmetri*.

”*No game should ever be decided by factors outside a player’s control. This has a simple logical consequence: The game has to be symmetrical at the level of significant game factors. Asymmetries, which are often necessary for reasons of realism or aesthetics, should be confined to the minor factors that cannot individually sway the outcome of the game.*”

*(*Rollings & Morris, 2004, s. 107)

Författarna nämner även kortfattat hur symmetri implementeras i designen:

”*Symmetry in design means that the choices available to all players are functionally the same. This is the easy way to make sure the game is fair. The hard way is to give each player different choices but to try to balance those choices so that every player has the same chance of success.*”

*(*Rollings & Morris, 2004, s. 109)

Rollings & Morris (2004) menar här alltså att symmetri innebär att de val spelare kan göra i slutändan resulterar i samma ”utfall”. Det enklaste sättet att göra detta på är att göra alla val tillgängliga till spelarna funktionellt likadana, det svåraste sättet är att tillåta funktionellt olika valmöjligheter men försöka balansera dessa så att samtliga spelare har lika stor chans att lyckas.

Vikten av realism i relation till balans diskuteras även:

”*Also remember that it’s only games that ought to be fair. A pure simulation may have the primary aim of authentically re-creating a historical situation, in which the question of balance between players may only be secondary.*”

*(*Rollings & Morris, 2004, s. 110)

Här menar alltså författarna att balans endast har stor vikt i den mer ”klassiska” sortens spel. Simulatorer (om än fortfarande spel) kan ha som huvudsakligt syfte att vara autentiska och återskapa realism och då kan det hända att balansen mellan spelare blir av sekundär natur.

5.4 Techniques for Achieving Play Balance

I denna artikel diskuterar författaren T. Cadwell (2008) olika tekniker för balansering av spel, bland annat *complexity control* och *macro/microcalibration*.

Complexity control beskriver Cadwell på följande sätt:

“*Complexity control can best be summarized as "keep it simple, stupid". Excessively complex game systems are harder to understand, and thus, harder to balance. An overcomplicated system usually is developed either because an initial design that was poor was endlessly "patched" with design fixes (leaving an incoherent mess that in theory works) or because of the all-too-ancient "too many cooks in the kitchen" phenomena, which often also corresponds to a lack of consistent design motivation. An added advantage to complexity control is that it avoids a number of potential gameplay issues. In particular, just as complex game systems are hard to understand and in turn to balance, they are also harder for players to understand, and after a certain point, to enjoy. An all too common design mistake is to favor complexity over depth, which leads to extreme play balance difficulties, and confusing and difficult to understand gameplay.*”

(T. Cadwell, 2008, s. 2)

Complexity control handlar alltså om att hålla den grundläggande designen av sitt spel på en ”enkel nivå”. Dålig ursprunglig design kan leda till att överdrivet komplicerade system skapas, vilket leder till att spelet blir mycket svårt att balansera men även att de blir svårare för spelarna att förstå (och i förlängningen – njuta av). Komplexitet bör aldrig ersätta djup!

Macrocalibration anser Cadwell vara det första steget i det praktiska balansarbetet och förklarar det på följande sätt:

"*Macrocalibration should always be completed before microcalibration is begun; small balance changes will be washed away and made into useless work if the foundation that the game rests on is still in transition. While macrocalibrating, the goal is to "find" the target gameplay that is described in the design document. Obviously, you can't polish the gameplay with small tweaks when you aren't even sure you've gotten the core gameplay to manifest!*

In order to zero in on the core gameplay desired, it is important to explicitly define what the core gameplay is like and how it will manifest. Once this is done, it is possible to establish some sort of baseline, or 'anchor' as Ensemble Studios calls it. For instance, for game pacing you might set a baseline of 'roughtly ten minute-long games' or for the player character's toughness you might set a baseline of "three attacks from a dangerous monster should be deadly". Once you get an incarnation of each game element (ONE map, ONE character class, ONE dialogue, etc) that works satisfactorily well, it is easy to expand upon the game, using the baseline game element as a ballpark figure to start from."

(T. Cadwell, 2008, s. 3)

Efter ovanstående process efterlyser Cadwell *microcalibration*, som beskrivs på följande sätt:

"*Once a game is macrocalibrated, game balance must be fine-tuned. If the game is at least somewhat fun, without any huge glaring problems, then it's probably macrocalibrated and ready for the small details. Microcalibration is the process by which the game designer applies small tweaks in order to perfect the state of balance of the game. A tweak can be defined as the set of changes such that the change in value is less than 10% for a "global" value (something that effects many game elements) and less than 30-40% for a "local" value (a particular game element).*

The biggest challenge in microcalibrating is simply identifying problems. Once a problem is identified, it is just a question of tweaking values slightly, and in such a way that one is not creating new problems. Good element modularity and pre-planning will really be a life-saver at this stage - without them, it may not be possible to balance the game in a reasonable time frame."

(T. Cadwell, 2008, s. 4)

Med makrokalibrering menar Cadwell alltså att stora förändringar i balansen bör göras först, för att säkerställa att den ”ursprungliga spelupplevelsen” förverkligas. Mindre ändringar som gjorts innan dessa riskerar att ”spolas bort” när större, mer övergripande ändringar görs.

Mikrokalibrering innebär enligt Cadwell att finjustera balansen i spelet, exempelvis genom att ändra på specifika värden på variabler. Är spelet relativt roligt är det förmodligen redan makrokalibrerat och därför redo för mikrokalibrering. Det svåraste med att mikrokalibera menar Tom är att faktiskt identifiera problemen. Han anser vidare att god översikt och planering är av största vikt för att spara tid vid mikrokalibrering.

6 Diskussion

Detta kapitel syftar till att diskutera det utförda arbetet i relation till de teorier som tagits upp, föra en kortare diskussion kring genus samt reflektera över lärdomar som dragits under arbetets gång.

6.1 Praktik i förhållande till teori

I detta avsnitt diskuteras de teorier som presenterats i föregående kapitel och de sätts även i relation till utfört arbete. Genom denna reflektion sätts de även i relation till uppsatsens syfte – att beskriva och diskutera de designmässiga beslut som tagits vid balanseringen, samt utvärdera dessa.

6.1.1 Patterns for Game Mastery and Balancing

Björk & Holopainen (2005) menar i grund och botten att spelarskicklighet alltid skall vara den avgörande faktorn för utgången av ett spel. Detta påstående stämmer väl överrens med den ursprungliga tanken bakom balanseringen av samtliga skadevärden:

”*Den grundläggande tanken är kort och gott att en skicklig spelare skall ha (i princip) lika goda möjligheter att vinna med båda stilarna, förutsatt att denne behärskar dem båda lika bra.*”

(se avsnitt Project Colosseum – s. 2)

Jag anser att detta påstående förverkligas praktiskt genom balanseringen av samtliga skadevärden. Uträkningarna har lett till att samtliga värden ”står i god relation” till varandra och således uppnås ett tillstånd där två spelare med likadana förutsättningar bör ha lika stor chans att besegra varandra, oavsett vilken stil de använder.

Det känns med andra ord ganska skönt att veta att vi var ”på rätt spår” med vår design redan från början. Själva teorin i sig känns i och för sig ganska självklar, en spontan tanke är att det finns väldigt få exempel på spel som (i praktiken) helt och hållet förlitar sig på slump, som även är roliga att spela, sett ur ett längre perspektiv.

Visst kan det vara roligt att singla slant några gånger ibland men inte är det väl något en person kan ha som sin ”största hobby”?

Gällande *preemptive* och *correcting* balance anser jag att det utförda arbetet mest liknar *preemptive* balance. Detta eftersom denna form av arbete mest syftar till att förebygga att obalans uppstår – varje attack har ett antal förbestämda värden som är balanserade gentemot varandra. Problematiken med det utförda arbetet är att det inte nödvändigtvis tar hänsyn till *correcting* balance. Balansen kan på ett sätt anses vara väldigt snäv, eftersom den till viss del förutsätter ett ”korrekt” spelande.

Å andra sidan så är inte *correcting* balance direkt applicerbart på just detta arbete, eftersom det helt enkelt inte tillhör den formen av balansering i grunden. Om en mer dynamisk formel för skadeändring beroende på spelares tillstånd skulle arbetas fram kan det eventuellt ses som en form av *correcting* balance. I grund och botten handlar det dock främst om att just detta arbete mer är av typen *preemptive* – förutsättningarna för alla spelare är likadana, oavsett stil.

6.1.2 The Internal Economy of Games and Game Balancing

Återigen diskuteras här spelarskicklighet som en avgörande faktor för framgång. På ett sätt kan det ses som att detta påstående står i stark kontrast till det utförda arbetet, där skickligheten hos varje spelare antagits vara likvärdigt för samtliga spelare (se Antaganden – s. 18).

Innebär detta i förlängningen att *Project Colosseum* balanserats på ett negativt sätt? Har information på något sätt misstolkats, vilket i förlängningen leder till en felaktig motivering av balans?

Jag anser att svaret på dessa frågor är ett rungande NEJ. Visserligen håller jag med fullständigt om att skicklighet skall vara den avgörande faktorn för resultatet men antagandet om likvärdig skicklighet anser jag vara helt och hållet nödvändigt för att arbetet skall kunna genomföras.

Vissa antaganden måste helt enkelt göras vid praktiskt balansarbete, annars blir antalet påverkande faktor alldeles för stort för att kunna ta hänsyn till i slutändan. Tar man inte hänsyn till så många faktorer som möjligt påverkas balansarbetet negativt. Det gäller alltså att göra *befogade* antaganden – antaganden som kan anses vara tillräckligt generella för att passa in på en stor del av spelets tilltänkta målgrupp.

Det är just detta som är ”kruxet” med detta balansarbete – under särskilda omständigheter fungerar det säkert alldeles utmärkt men majoriteten av spelarna kommer med största sannolikhet inte ens ”märka” det.

Jag anser att en av *Project Colosseums* absolut största styrkor är dess ”linjära balans”. För två spelare som inte använder sig av taktisk finess blir skillnaderna i skadevärden mycket små och detsamma gäller även (rent matematiskt) för ett mer avancerat spelande.

Huruvida detta fungerar till perfektion i praktiken eller inte låter jag vara osagt, det finns absolut en möjlighet att samtliga värden kommer att behöva omarbetas. Precis som Rollings & Adams (2003) nämner så är det i praktiken omöjligt att skapa ”perfekta” ekvationer för balansering. Detta beror delvis på att balansering inte alltid beror på numeriska värden (det kan exempelvis röra sig om areor och ytor istället) samt att antalet påverkande variabler har en tendens att bli *väldigt* stort i slutändan, oavsett hur förutseende designern tror sig vara i spelets första skeden.

Det går visserligen att argumentera för att all data i slutändan kan abstraheras till numeriska värden men detta är något som Rollings & Adams (2003) varnar för. Mänsklig finess och ”känsla” anses vara nödvändigt för att balansen skall bli riktigt bra. Jag håller verkligen med dessa herrar. Ett praktiskt exempel på detta i balansarbetet är bestämmandet av procentuella värden för varje attacktyp samt skadeökningen för varje ”steg” i varje combo. Arbetet hade visserligen sin grund i siffror och matematik (om än undermedveten) men först och främst bestämde jag dessa värden efter vad som ”kändes rätt”.

Jag tror även det är viktigt att påpeka att ovanstående påstående stämmer väl endast om det används med måtta. Rollings & Adams (2003) antyder även detta när de diskuterar vikten av eftertanke vid tweaking av variabler. ”Fingertoppskänsla” är således viktigt för balansering, så

länge det stannar vid just detta – ”instinkt” kan aldrig ersätta rationellt beslutsfattande helt och hållet.
Vidare nämner Rollings & Adams (2003) vikten av estetik framför ”perfekt matematik”. Även detta stämmer väl överrens med balansarbetet, eftersom samtliga combos i grund och botten utformats efter vår animatörs estetiska åsikter (se avsnitt Splits – s. 15). Detta kan tyckas vara ett ”bakvänt” sätt att arbeta men jag anser det vara väldigt svårt, om inte omöjligt att arbeta annorlunda.

I mina ögon får den konstnärliga friheten aldrig kvävas helt och hållet till förmån för spelmekanik i dagens spel. Denna åsikt grundar sig främst i marknadsmässiga funderingar.

Jag anser att dagens spelare är väldigt bortskämda på avseendet ”grafisk finess”. Tyvärr ser verkligheten ut såhär idag, tror jag. Det spelar ingen roll hur bra ett spels mekanik är, tycker spelarna att det är fult så kommer de inte att spela det! En återkoppling kan även göras till diskussionen om genus – det som är allt för annorlunda döms ut med en gång. Tragiskt men sant!

Naturligtvis skall man inte oroa sig för mycket om vad andra tycker men den slutgiltiga målsättningen är naturligtvis att lyckas lansera sin produkt. Detta innebär att ett visst fokus kring marknadens värderingar och åsikter *alltid* måste finnas och därmed styr spelutvecklingen.

I slutändan innebär alltså detta att ”perfekt” balans är lite av en utopi, eftersom det i praktiken verkar omöjligt att sammanfoga lysande grafik med fantastisk spelmekanik. Varje spel har sina brister. Oavsett om de är små eller stora så är det viktigaste att spelaren har roligt, sett till helheten!

Gällande *statisk* och *dynamisk* balans vill jag påstå att det utförda arbetet är av typen *statisk* balans. Just denna term påminner till stor del om *preemptive* balance, eftersom båda typer ser grundläggande mekanik som definierats från början. I *Project Colosseum* realiseras detta praktiskt i form av faktorer som angivna skadevärden, förmågan hos swings att dubblera skada, och så vidare.

Den *dynamiska* balansen är således jämförbar med *correcting* balance, eftersom detta är något som sker under spelets gång. Jag tror att balans i *Project Colosseum* till viss del uppstår som ett resultat av spelarnas egna ansträngningar. Det finns vissa givna förutsättningar men hur och när spelarna använder sig av dessa är helt och hållet upp till dem! Att ”tvinga” balans på spelare känns bara naivt eftersom det begränsar friheten hos dem.

Hellre frihet än fullständig jämlikhet i samtliga spelsituationer!

6.1.3 Player/Player Balance

Återigen nämns spelarskicklighet som den avgörande faktorn i ett välbalanserat spel, denna gång av herrarna Rollings & Morris (2004). De för denna diskussion vidare genom att ta upp principen med symmetri – resultatet av spelet skall aldrig avgöras på grund av faktorer utanför spelarens kontroll.

Just detta påstående anser jag i mångt och mycket validerar det utförda arbetet. Varje attack har jämförts gentemot sina motsvarigheter både internt och externt – de faktorer som spelarna själva styr över (attackerna) ger samma funktionella resultat (värdena är balanserade).

Rollings & Morris (2004) menar vidare att den svåraste formen av symmetri är att tillåta varje spelare att göra olika men balanserade val. Resultatet skall vara detsamma men utförandet kan tänkas skilja sig åt.

Detta bekräftas av den estetiska skillnaden mellan stilarnas olika combos. Förutom att de faktiskt ser annorlunda ut skiljer sig mappningen åt för vissa av dem. Det har ansetts nödvändigt att bibehålla viss ”igenkänning” vid mappning av input, för att inte inlärningen av samtliga attacker skall bli för svår för spelare.

I grund och botten anser jag det handla om att presentera spelaren för valmöjligheter som funktionellt är varandras motsvarigheter men samtidigt ”känns tillräckligt olika”. Detta anser jag uppfylls väl av balanserade skadevärden och ett brett estetiskt fokus.

Estetik och mekanik kompletterar således varandra väl, där syftet är att bibehålla spelarens ”illusion” av ett väl fungerande system.

Rollings & Morris (2004) tar vidare upp en (i mina ögon) extremt viktig punkt att ta hänsyn till vid balansering – spel är inte simulatorer!

Som nämnts i tidigare avsnitt anser jag ”perfekt” balans vara något av en utopi. Återigen kan man tala om att skapa en illusion för spelaren. Detta innebär att det praktiska balanseringsarbetet är något ”bakomliggande”. Arbetet skall sedan ”kännas” tillräckligt trovärdigt för att spelaren skall acceptera detta i majoriteten av fallen.

Det viktigaste är således att inget i spelet känns ”konstigt”, vilket jag anser att balanserade skadevärden har mycket stor inverkan på.

6.1.4 Techniques for Achieving Play Balance

T. Cadwell (2008) diskuterar vikten av att ha så låg komplexitet som möjligt på de system som skall balanseras. Det kan argumenteras för att den ekvationen för att balansera skadevärden i *Project Colosseum* som denna uppsats behandlar är alldeles för komplex. Vidare kan *Toolkit* tyckas vara svårförståelig och det går att diskutera huruvida dess struktur bör designas om.

Har abstraktionen av skadevärden tagits för långt? Har processen att balansera skada vuxit till ett ”monster”, som är omöjligt att helt och hållet styra över i slutändan?

På dessa frågor blir mitt svar ett bestämt NEJ. Vid första anblicken kan arbetet tyckas svåröverskådligt och svårförståeligt men detta anser jag inte beror på själva ekvationen. Det handlar snarare om att själva processen är väldigt lång och sker i flera steg, vilket säkert kan tänkas förvirra vid första anblicken. Samtliga uträkningar baseras dock på de allra enklaste formerna av matematik – addition, subtraktion, division och multiplikation.

Visserligen kan det avgöras att den långa processen är ett resultat av för hög komplexitet i spelets ursprungliga design. Till viss del håller jag med i detta påstående men har svårt att avgöra hur spelet hade kunnat designas annorlunda. Den komplexitet som finns i designen är till för att säkerställa ett visst djup i *Project Colosseum* och öka dess ”livslängd”.

Möjligen har vi i designen övervärderat dessa behov men samtidigt är spelets komplexitet något som jag anser växer fram efterhand. Vid spelutveckling är det mest vitala steget att skapa en enkel, fungerande och framför allt *rolig* prototyp av spelets grundmekanik så fort som möjligt. Denna prototyp växer sedan i storlek allteftersom nya idéer och tankar presenteras.

Även detta anser jag vara en utopi vid spelutveckling – som designer har man inte alltid svar på allt! Man kan inte ensam lastas som ansvarig för *all* kreativitet, dock är det ens ansvar att vara öppen för *samtliga* idéer som presenteras för en.

När det gäller *mikro*/*makrokalibrering* så uppenbarar sig balansarbetets överlägset största brist. *Mikrokalibering* har till viss del skett utan att ta hänsyn till *makrokalibrering*. Detta bevisas rent praktiskt av att skillnaderna i antalet effektiva key-frames för combos av olika typ i vissa fall är mycket små. Eftersom skillnaderna i sammanlagda skadevärden dock skiljer sig stort innebär detta att en form av obalans uppstår.

Detta eftersom vissa attacker helt förlorar sitt syfte. Det finns inget som motiverar en spelare att utföra exempelvis en *Light combo* om denne tar lika lång tid (har lika stort antal effektiva key-frames) som en *Heavy combo*, eftersom skadevärdet är mycket högre för en *Heavy combo*.

Lösningen på detta problem enligt T. Cadwell (2008) blir således att omarbeta *makrokalibreringen*. I praktiken innebär det alltså att vår animatör eventuellt får tvingas göra om ett antal animationer, där det huvudsakliga syftet är att lägga till eller ta bort effektiva key-frames.

Detta problem påvisar en viktig modifikation av påståendet som rör estetikens vikt: Grundläggande riktlinjer bör alltid upprätthållas som ett resultat av *makrokalibreringen*!

6.2 Genus – alltid aktuellt?

Genusperspektivet är en ständigt aktuell fråga i dagens samhälle och den känns särskilt relevant och viktig inom kontexten dataspelsutveckling i stort, med tanke på den traditionella ”mansdominansen” i branschen.

För just denna uppsats finner jag det tyvärr svårt att dra några större slutsatser om genusperspektiv. Detta eftersom jag anser arbetet i sig vara så pass abstrakt att en konkret diskussion kring genus och könsroller mest känns aningen krystat.

Jag anser kärnan i mitt arbete vara beskrivningar av det matematiska arbete som ligger till grund för balansen i *Project Colosseum*. I detta sifferbaserade sammanhang anser jag inte riktigt att genusperspektivet ”hör hemma”, särskilt med tanke på att vi i den ursprungliga designen valt att fokusera enbart på estetik.

Olika vapen och rustningar påverkar således inte spelmekaniken och detsamma gäller för valet av kön. Vi har inte velat göra några absoluta gränsdragningar och därför beslutat att utrustning och kön endast skall fungera som ”utsmyckning”.

Det skulle naturligtvis vara möjligt att applicera någon form av design där olika aspekter av genus tas i beaktande. Det mest konkreta för denna uppsats skulle antagligen vara att diskutera huruvida skadevärden påverkas av kön.

Gör en kvinnlig gladiator lika mycket skada med sina attacker som en manlig? Har attackerna samma hastighet?

Ett tillförande av dessa faktorer skulle innebära att allt balansarbete skulle behöva göras om, eftersom helt nya variabler förs in i ekvationen. Dessutom är det inte längre säkert att ekvationen skulle kunna hantera samtliga dessa nya variabler utan att först omarbetas.

I förlängningen kan man absolut ha genus i åtanke under hela utvecklingen och jag anser att det är ett mycket intressant sätt att motivera och främja införande av dynamik i spelsystem. Detta eftersom genus kan appliceras på i stort sett allt i mina ögon: hälsa, skada, styrning, etc.

Jag anser det dock finnas en viss risk med att ”göra för stor sak” av genusperspektivet. Samtliga diskussioner och funderingar måste alltid sättas i relation till varandra och deras absoluta syfte/nytta bör ständigt diskuteras. Genom att sätta manligt och kvinnligt i stor kontrast till varandra riskerar man att underminera främjandet av jämställdhet. Detta anser jag tyvärr vara särskilt sant för det svenska samhället i synnerhet, där ”annorlunda” allt för ofta är att betrakta som något negativt.

6.3 Reflektioner kring utfört arbete

Det har funnits två målsättningar med detta arbete: den *professionella* och den *individuella*.

Den *professionella* målsättningen har varit att upprätta en form av balans i *Project Colosseum*, något jag anser har uppnåtts med ovanstående arbete. Balansen i sig är långt ifrån perfekt men det nuvarande resultatet känns som en mycket stabil grund att ”bygga vidare” på.

Den *individuella* målsättningen har varit att få en större förståelse för hur balansering fungerar samt att arbeta praktiskt med detta så mycket som möjligt. Båda dessa punkter anser jag uppfyllts väl under detta examensarbete, särskilt den praktiska biten.

Gällande teorin så anser jag att just balansering är ett så brett och abstrakt begrepp att det omöjliggör exakta beskrivningar av lösningar. Spontant känner jag att det är just detta som är tjusningen med balansering, att det aldrig finns en ”given” lösning på ett problem. Balansering lämnar således mycket åt designerns kreativitet, något jag personligen värdesätter väldigt högt.

Jag anser även att mina reflektioner kring ovanstående teorier visar på en genomgående kunskapsprocess. Även om ämnet balansering är tämligen abstrakt anser jag mig ha kommit fram till ett flertal goda observationer. Jag tycker mig även ha lyckats tämligen väl med att realisera abstrakta termer och teorier till praktiska jämförelser och observationer, något som enligt mig visar på god kunskap och förståelse.

6.3.1 Positivt

- Ett väl genomtänkt matematiskt arbete har utförts.
- Samtliga aspekter av processen har förklarats.
- Numerisk balans (i någon form) existerar.
- Djupare tankar kring design och utveckling av balans har uppkommit.
- Djupare insikt i praktiskt balanseringsarbete och dess svårigheter har uppnåtts.
- Arbetsmetodiken har anpassats väl utifrån rådande begränsningar hos verktyg och applikationer.
- Vikten av riktvärden vid praktiskt balanseringsarbete har förtydligats.
- Djupare kunskap om generella teorier har införskaffats.

6.3.2 Negativt

- Arbetet kan tänkas svårförståeligt och komplicerat för utomstående.
- Tydliga brister finns i makrokalibreringen – balansen brister internt, i relationen mellan skada och antal effektiva key-frames för vissa combos av olika typ.
- Antalet faktorer att ta hänsyn till har varit väldigt stort och arbetet kan därmed tyckas svåröverskådligt.
- Balansarbetet tar ej hänsyn till samtliga faktorer i *Project Colosseum* (exempelvis input, attackers areor).
- Beskrivning av processer behöver kanske göras mer ”användarvänligt” – exempelvis genom användandet av bilder.

6.3.3 Slutsats

Att balansera vilket spel som helst är en mycket svår och tidskrävande uppgift. Denna uppsats har förhoppningsvis gett läsaren en viss insikt i hur praktiskt balanseringsarbete kan gå till. Teoretisk balansering handlar om att ta hänsyn till så många olika faktorer som möjligt men detta garanterar bara balans inom en viss (begränsad) kontext. Praktisk balansering handlar dock om att ta hänsyn till såväl estetik som mekanik, där kärnan i god balans kan ses som ett upprätthållande av den ”fungerande illusionen”.

Referenser

Litteratur

Björk, S. & Holopainen, J. (2005) *Patterns in game design.* Massachusetts: Charles river media.

Rollings, A & Adams, E. (2003) O*n Game Design.* United States of America: New Riders Publishing.

Rollings, A & Morris, D. (2003) *Game Architecture and Design: A New Edition.* United States of America: New Riders Publishing.

Internet

Cadwell, T. (2008) *Techniques for Achieving Play Balance*. GameDev.net [http://www.gamedev.net/reference/design/features/balance/default.asp] (2008-05-06)

Gothia Science Park [http://www.gsp.se/templates/start.php?top=0&me=0&sub=0] (2008-05-06)

Högskolan i Skövde [http://www.his.se/] (2008-05-06)

The Game Incubator [http://www.gameincubator.se/] (2008-05-06)

Spel

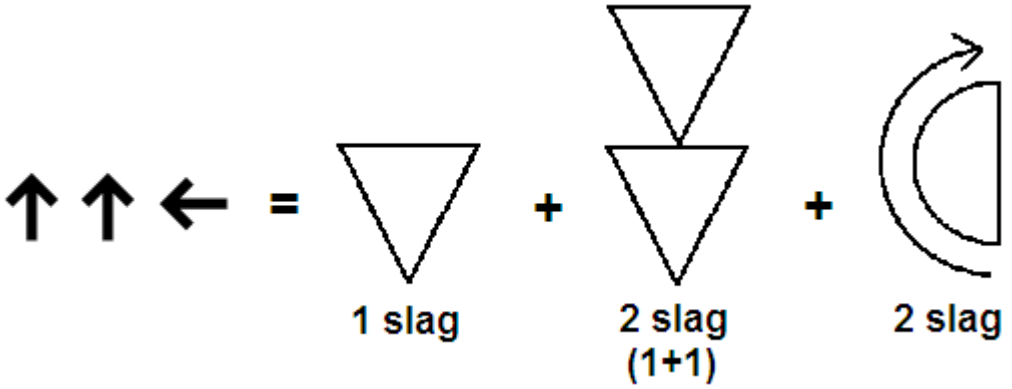
Soul Calibur III (2005) Namco

Bilagor

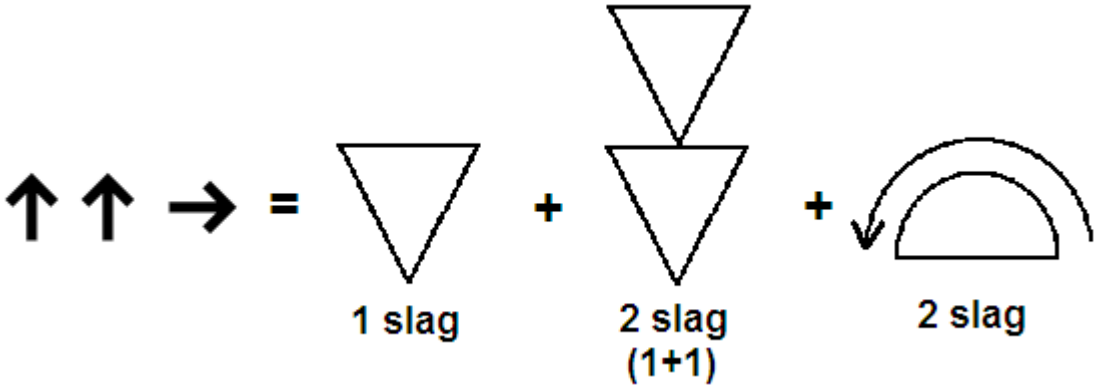
Bilaga 1 – Combo design (Dual-Wield)

Light:

Combo 1

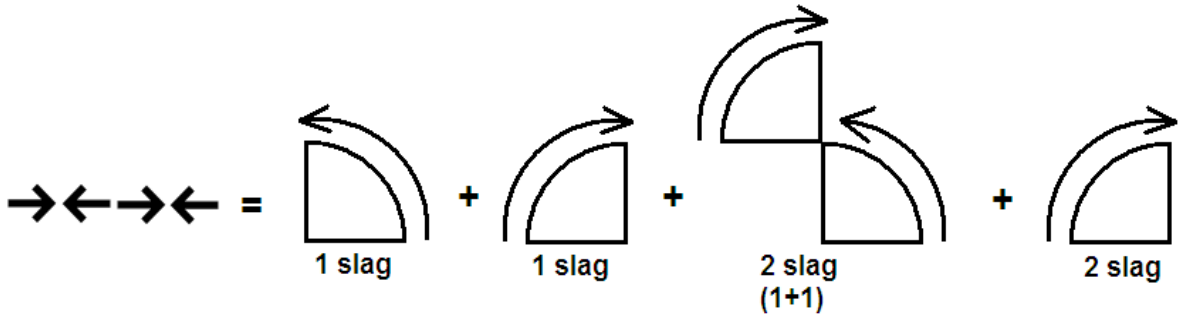


Combo 2



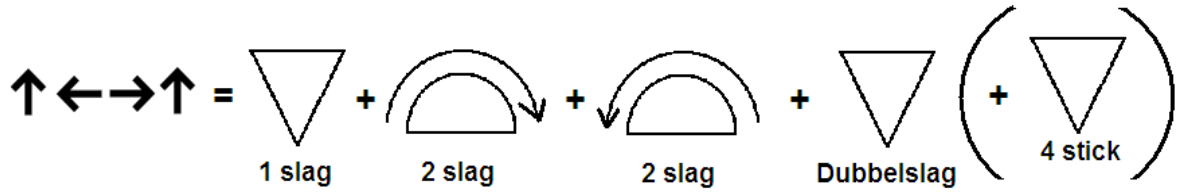
Heavy:

Combo 3

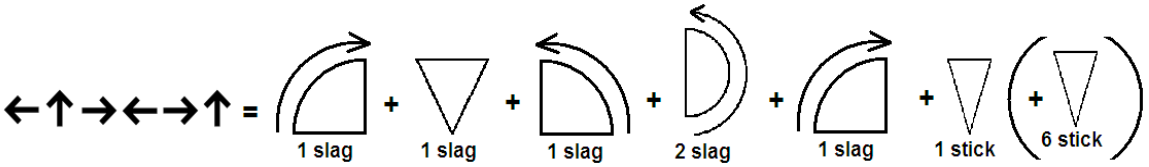


Advanced:

Combo 4

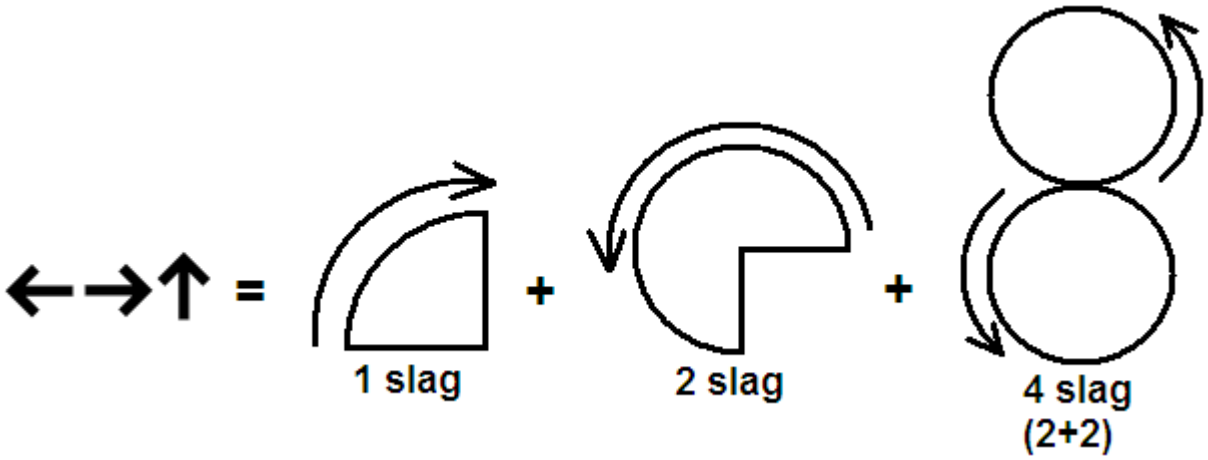


Combo 5



AOE:

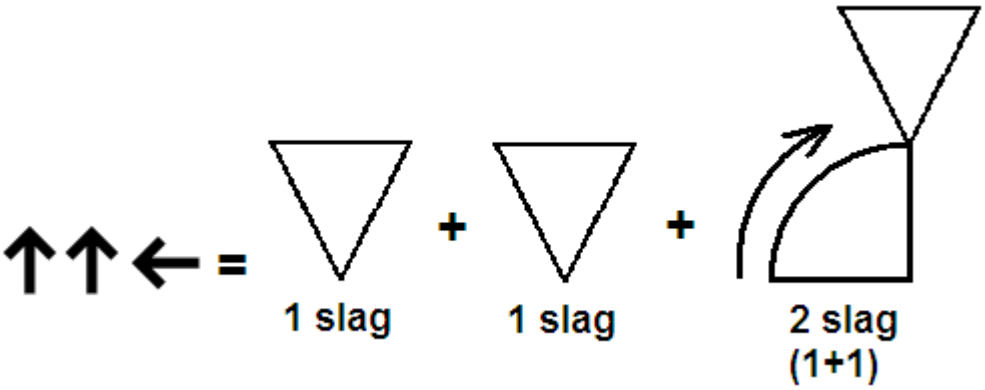
Combo 6



Bilaga 2 – Combo design (Mace)

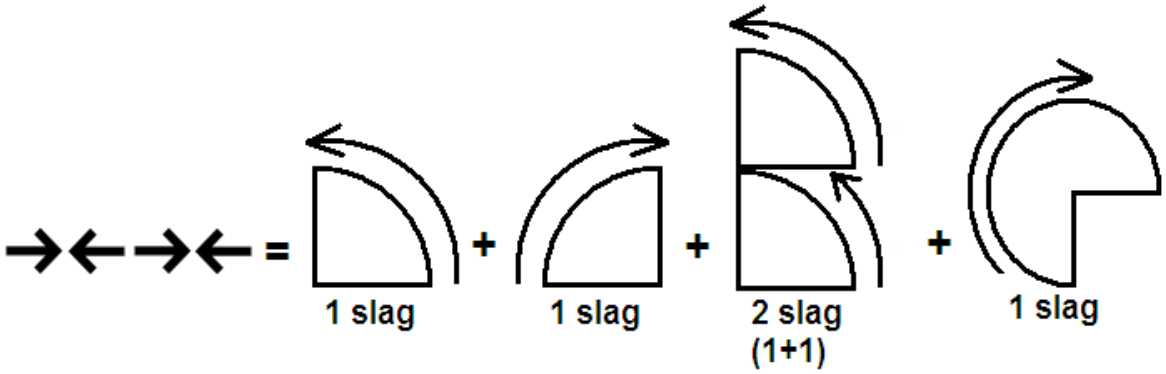
Light:

Combo 1

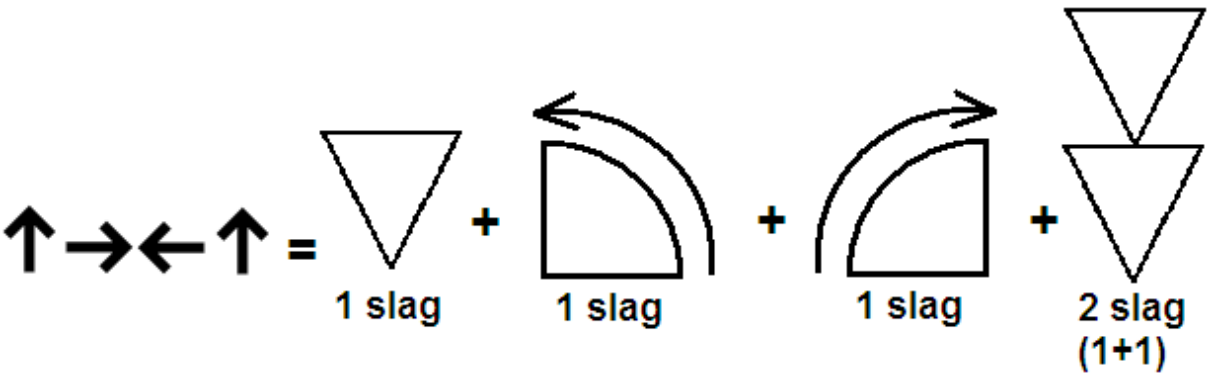


Heavy:

Combo 2

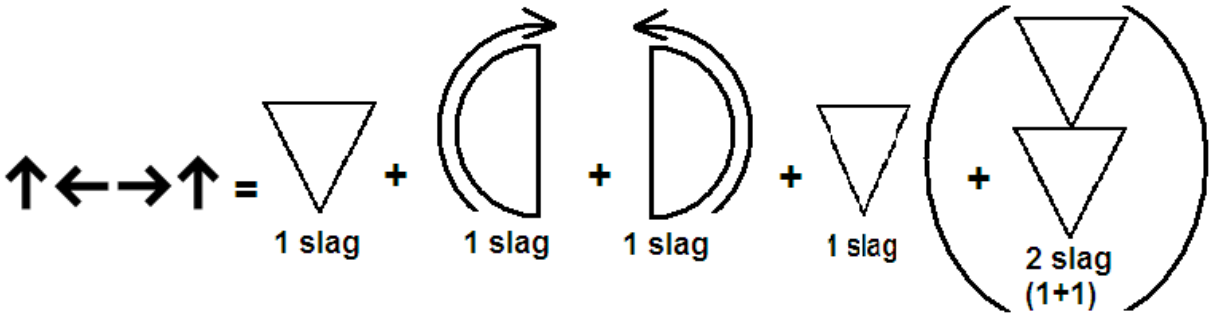


Combo 3

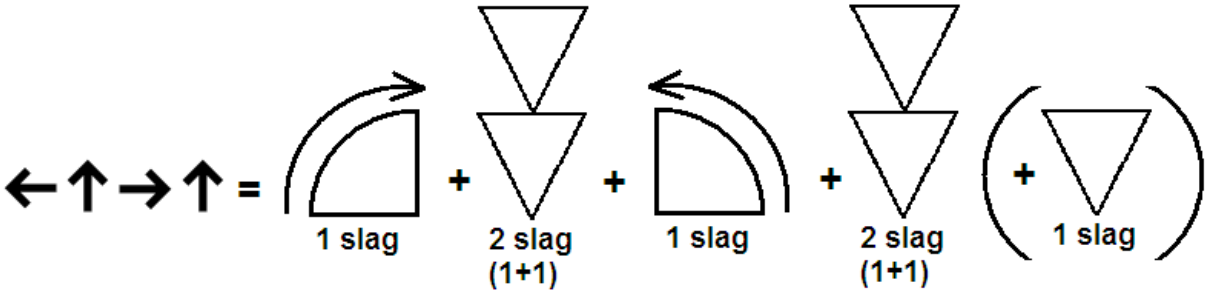


Advanced:

Combo 4

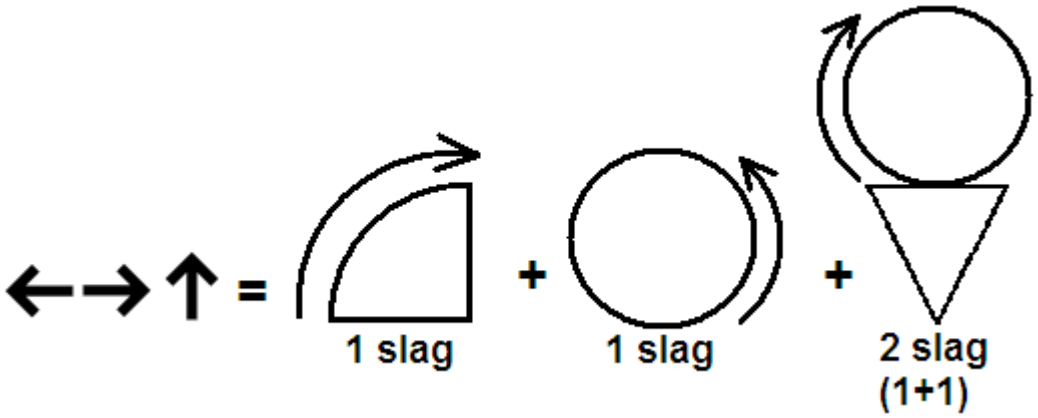


Combo 5



AOE:

Combo 6



Bilaga 3 – Balanserade skadevärden

Mace – Combo 1 (Light)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Forward)	**18**	**1**	18 / 1 = **18**	**N**	18 / 1 = **18**
Steg 2 (Forward)	**37**	**1**	37 / 1 = **37**	**N**	37 / 1 = **37**
Steg 3 (Left)	**50**	**2**	50 / 2 = **25**	**J**	25 / 2 = **13**

Dual-Wield – Combo 1 (Light)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Forward)	**14**	**2**	14 / 2 = **7**	**N**	7 / 1 = **7**
Steg 2 (Forward)	**24**	**2**	24 / 2 = **12**	**N**	12 / 1 = **12**
Steg 3 (Left)	**43**	**2**	43 / 2 = **21,5**	**J**	21,5 / 2 = **11**

Dual-Wield – Combo 2 (Light)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Forward)	**14**	**2**	14 / 2 = **7**	**N**	7 / 1 = **7**
Steg 2 (Forward)	**13**	**2**	13 / 2 = **6,5**	**N**	6,5 / 1 = **7**
Steg 3 (Right)	**31**	**2**	31 / 2 = **25,5**	**J**	25,5 / 2 = **13**

Mace – Combo 2 (Heavy)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Right)	**16**	**1**	16 / 1 = **16**	**J**	16 / 2 = **8**
Steg 2 (Left)	**21**	**1**	21 / 1 = **21**	**J**	21 / 2 = **11**
Steg 3 (Right)	**40**	**2**	40 / 2 = **20**	**J**	20 / 2 = **10**
Steg 4 (Left)	**75**	**1**	75 / 1 = **75**	**J**	75 / 2 = **38**

Mace – Combo 3 (Heavy)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Forward)	**18**	**1**	18 / 1 = **18**	**N**	18 / 1 = **18**
Steg 2 (Right)	**30**	**1**	30 / 1 = **30**	**J**	30 / 2 = **15**
Steg 3 (Left)	**52**	**1**	52 / 1 = **52**	**J**	52 / 2 = **26**
Steg 4 (Forward)	**75**	**2**	75 / 2 = **37,5**	**N**	37,5 / 1 = **38**

Dual-Wield – Combo 3 (Heavy)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Right)	**11**	**1**	11 / 1 = **11**	**J**	11 / 2 = **6**
Steg 2 (Left)	**31**	**1**	31 / 1 = **31**	**J**	31 / 2 = **16**
Steg 3 (Right)	**52**	**3**	52 / 3 = **17,3**	**J**	17,3 / 2 = **9**
Steg 4 (Left)	**72**	**2**	72 / 2 = **36**	**J**	36 / 2 = **18**

OBS!
Dual-Wield jämförs här gentemot snittvärdet för totala skadan på båda Heavy combos → (152 + 175) / 2 = **164**.
Detsamma gäller även för effektiva key-frames, där snittvärdet blir → (56 + 58) / 2 = **57**.

Mace – Combo 4 (Advanced)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**18**	**1**	18 / 1 = **18**	**J**	18 / 2 = **9**
Steg 2 (Forward)	**11**	**1**	11 / 1 = **11**	**N**	11 / 1 = **11**
Steg 3 (Right)	**12**	**2**	12 / 2 = **6**	**J**	6 / 2 = **3**
Steg 4 (Forward)	**13**	**1**	13 / 1 = **13**	**N**	13 / 1 = **13**

Dual-Wield – Combo 4 (Advanced)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**14**	**1**	14 / 1 = **14**	**J**	14 / 2 = **7**
Steg 2 (Forward)	**11,3**	**3**	11,3 / 3 = **3,8**	**N**	3,8 / 1 = **4**
Steg 3 (Right)	**12,3**	**2**	12,3 / 2 = **6,2**	**J**	6,2 / 2 = **3**
Steg 4 (Left)	**14,3**	**2**	14,3 / 2 = **7,2**	**J**	7,2 / 2 = **4**

Mace – Combo 6 (AOE)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**16**	**1**	16 / 1 = **16**	**J**	16 / 2 = **8**
Steg 2 (Right)	**27**	**2**	27 / 2 = **13,5**	**J**	13,5 / 2 = **7**
Steg 3 (Forward)	**45**	**2**	45 / 2 = **22,5**	**N**	22,5 / 1 = **23**

Dual-Wield – Combo 6 (AOE)

Steg	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Steg 1 (Left)	**11**	**1**	11 / 1 = **11**	**J**	11 / 2 = **6**
Steg 2 (Right)	**33,5**	**2**	33,5 / 2 = **16,8**	**J**	16,8 / 2 = **8**
Steg 3 (Forward)	**52,5**	**3**	52,5 / 3 = **17,5**	**N**	17,5 / 1 = **18**

Övriga attacker

Attacktyp	Totalt antal effektiva key-frames	Relation (jämfört med riktvärdet)	Uppdaterat skadevärde
*Charge attack – Left (Mace – **riktvärde**)*	**40**	**-**	**52**
Charge attack – Left (Dual-Wield)	**22**	22 / 40 = **0,55**	0,55 x 52 = **29**
*Charge attack – Right (Mace – **riktvärde**)*	**41**	**-**	**53**
Charge attack – Right (Dual-Wield)	**23**	23 / 41 = **0,56**	0,56 x 53 = **30**
Charge attack – Forward (Mace)	**19**	19 / 21 = **0,9**	**24**
*Charge attack – Forward (Dual-Wield – **riktvärde**)*	**21**	**-**	1,1 ((1-0,9) + 1) x 24 = **26**
*Combo 4 – Special attack (Mace – **riktvärde**)*	**51**	**-**	**210**
Combo 4 – Special attack (Dual-Wield)	**34**	34 / 51 = **0,7**	0,7 x 210 = **147**
Combo 5 – Special attack (Mace)	**50**	50 / 52 = **0,96**	**206**
*Combo 5 – Special attack (Dual-Wield – **riktvärde**)*	**52**	**-**	1,04 ((1-0,96) + 1) x 206 = **214**
*Rage attack (Mace – **riktvärde**)*	**50**	**-**	**175**
Rage attack (Dual-Wield)	**39**	39 / 50 = **0,8**	0,8 x 175 = **140**

Attacktyp	Balanserat skadevärde	Antal kollisioner	Uppdaterat värde	Swing? (J/N)	Slutgiltigt värde
Charge attack – Left (Mace)	**52**	**2**	52 / 2 = **26**	**J**	26 / 2 = **13**
Charge attack – Left (Dual-Wield)	**29**	**2**	29 / 2 = **14,5**	**J**	14,5 / 2 = **7**
Charge attack – Right (Mace)	**53**	**3**	53 / 3 = **17,7**	**J**	17,7 / 2 = **9**
Charge attack – Right (Dual-Wield)	**30**	**3**	30 / 3 = **10**	**J**	10 / 2 = **5**
Charge attack – Forward (Mace)	**24**	**1**	24 / 1 = **24**	**J**	24 / 2 = **12**
Charge attack – Forward (Dual-Wield)	**26**	**2**	26 / 2 = **13**	**J**	13 / 2 = **7**
Combo 4 – Special attack (Mace)	**210**	**2**	210 / 2 = **110**	**N**	110 / 1 = **110**
Combo 4 – Special attack (Dual-Wield)	**147**	**4**	147 / 4 = **36,8**	**N**	36,8 / 1 = **37**
Combo 5 – Special attack (Mace)	**206**	**1**	206 / 1 = **206**	**N**	206 / 1 = **206**
Combo 5 – Special attack (Dual-Wield)	**214**	**5**	214 / 5 = **42,8**	**N**	42,8 / 1 = **43**
Rage attack (Mace)	**175**	**3**	175 / 3 = **58,3**	**N**	58,3 / 1 = **58**
Rage attack (Dual-Wield)	**140**	**2**	140 / 2 = **70**	**N**	70 / 1 = **70**