

Kristian Kallio

# Pelituntuman suunnittelu ja toteutus

## 2D-toimintapeleissä

A pixelated landscape featuring a range of mountains in shades of brown and grey. The word "OODA" is written in a large, yellow, pixelated font across the middle of the scene. The background is a dark, textured gradient.

OODA

Tradenomi  
Tietojenkäsittely  
Syksy 2020



KAMK • University  
of Applied Sciences

## Tiivistelmä

**Tekijä:** Kallio Kristian

**Työn nimi:** Pelituntuman suunnittelu ja toteutus 2D toimintapeleissä

**Tutkintonimike:** Tietojenkäsittelyn tradenomi

**Asiasanat:** pelituntuma, pelisuunnittelu, juissi, öljy

Tässä opinnäytetyössä selvitettiin pelituntuman roolia videopeleissä ja niiden kehityksessä. Työssä määriteltiin pelituntuma, juissi ja öljy käsitteinä sekä pelituntuman mittaaminen käyttämällä pelituntuman metriikoita.

Pelituntumalle tarpeellisiksi elementeiksi todettiin reaaliaikainen kontrolli, spatiaalinen simulaatio ja viihteisyys. Pelituntumaa voidaan myös mitata, vaikka mitattaessa metriikat ovatkin usein subjektiivisia. Pelituntuman metriikoiksi määriteltiin syöte, vaste, konteksti, metafora, viihteisyys, säännöt ja yhtenäisyys. Juissin luokiteltiin olevan mitattavissa viihteistyynä, kun taas öljy liittyy eniten vasteeseen ja kontekstiin.

Pelituntuman todettiin vaikuttavan pelaamiseen lähinnä hetkittäisessä pelaamisessa, mutta voi myös sääntöjen kautta vaikuttaa pelaajan pitkäaikaiseen motivaatioon. Juissi vahvistaa pelin palautetta pelaajan syötteeseen tai tilan vaihtumiseen audiovisuaalisilla efekteillä, kun taas öljy minimoi pelaajan vaivaa syötteiden tekemiseen ja auttaa peliä tulkitsemaan syötteiden haluttuja tuloksia joustavammin.

Työssä tehtiin tasohyppely ja paikallinen moninpeli. Peleihin suunniteltiin ja implementoitiin tehokeinoja, joilla pelituntumaa parannettiin. Käytetyt tehokeinot keskittyivät tasohyppelyssä eniten pelaajan syötteen luotettavampaan tulkitsemiseen ja intuitiivisempaan tasosuunnitteluun, kun taas monipelissä pelituntumaa parannettiin lähinnä juissaamisella.

## **Abstract**

**Author:** Kallio Kristian

**Title of the Publication:** Designing and Implementing Game Feel in 2D Action Games

**Degree Title:** Bachelor of Business Administration

**Keywords:** game feel, game design, juice, oil

The subject of the thesis was to analyse game feel in video games and game development. Juice and oil were also defined as terms relating to game feel. Game feel was determined to be measurable by mainly soft metrics.

Real-time control, spatial simulation and polish were deemed necessary for game feel to exist. Game feel can also be measured through metrics, although they were deemed mostly subjective. Defined metrics are input, response, context, metaphor, polish, rules, and consistency. Juice was determined to be measurable through polish while oil was linked primarily to response and context.

Game feel was proved to impact mostly the moment-to-moment gameplay, although rules can also affect the player's motivation in the long run. Juice reinforces the feedback that games give to inputs and changing of states by the use of audiovisual effects, while oil minimizes player effort when making inputs and helps the game interpret the wanted response for the input more flexibly.

A platformer and a local multiplayer game were made as a part of this thesis. Game feel was improved by implementing means specific to both genres of the games. In the platformer, game feel was improved mainly by making the player's input more reliably interpreted via the use of oil and making level design more intuitive. Juice was the primary means of improving game feel in the local multiplayer game.

## Sisällys

1	Johdanto .....	1
2	Pelituntuman määrittely .....	2
2.1	Reaaliaikainen kontrolli .....	2
2.2	Spatiaalinen simulaatio .....	4
2.3	Viimeistely .....	4
3	Pelituntuman metriikat .....	6
3.1	Syöte ja vaste.....	6
3.2	Konteksti.....	8
3.3	Metafora ja viimeistely.....	10
3.4	Säännöt.....	13
3.5	Yhtenäisyys.....	14
4	Juussin määrittely .....	15
5	Öljyn määrittely .....	18
6	Tasohyppely .....	21
6.1	Syötepuskurin hyödyntäminen .....	22
6.2	Tapahtumien asteittaisuus.....	25
6.3	Törmäysalueiden suunnittelu.....	27
6.4	Pelihahmon attribuutit ja tasosuunnittelu.....	34
6.5	Pelihahmon viimeistely .....	37
7	Paikallinen moninpeli .....	40
7.1	Yksinkertaisen syötteen kääntäminen monimutkaiseksi vasteeksi .....	40
7.1.1	Ohjaussauva .....	41
7.1.2	Painike .....	43
7.2	Audiovisuaaliset efektit kommunikaationa.....	45
7.3	Kameran tehtävä ja kameraefektien tehokeinot .....	50
7.3.1	Käytetyt kameraefektit .....	51
7.4	Luettavuuden säilyttäminen .....	56
8	Pohdinta .....	58
9	Yhteenveto .....	61

Lähteet

## Termiluettelo

Interpolaatio	Kahden arvon väliin sijoittuvan arvon laskeminen
Juissi	Mikä tahansa pelin sisäinen audiovisuaalinen efekti
Kehys	Yksittäinen hetki, jonka peli piirtää ruudulle
Kehystaajuus	Kuinka monta kehystä peli näyttää sekunnissa
Muuttuja	Pelin simulaatiossa muutettava arvo
Pelihahmo	Pelin sisällä oleva hahmo, jota pelaaja ohjaa
Peliobjekti	Esine tai asia pelimaailmassa
Verbi	Mikä tahansa toiminta, jonka pelihahmo voi suorittaa
Öljy	Mekaniikat, jotka pienentävät pelaajan vaivaa pelatessa

## 1 Johdanto

Kaksi mekaanisesti samankaltaista peliä voivat tuntua pelaajalle hyvin erilaiselta pelatessa. Pelin visuaalisen ulkoannin lisäksi konepellin alla toimivat osat määrittävät, miltä pelaajan suorittamat toiminnot tuntuvat. Opinnäytetyön idea syntyi halusta selvittää, miksi kaksi näennäisesti samankaltaista systeemiä voivat tuntua hyvinkin erilaisilta. Ideana oli selvittää, mitkä nyanssit muodostavat hyvätuntuisen pelihahmon verbin, ja tutkia näiden nyanssien takana olevaa teoriaa ja linkittää verbit siihen.

Opinnäytetyössä kerrotaan, mistä osista pelituntuma muodostuu yhden teorian mukaan ja miten juissi ja öljy käsitteinä liittyvät siihen. Työssä selvitetään myös, miten analysoida ja mitata pelituntumaa käyttämällä eri pelituntuman metriikoita. Lopuksi pelituntuman toteutusta havainnollistetaan konkreettisesti tutkimalla kahta eri peliprojektia. Toteutuskeinoja selitetään osittain teknisellä tasolla sekä keinojen alla piilevällä teoriatasolla.

Tavoitteena on parantaa varsinkin uusien pelinkehittäjien tietämystä aiheesta ja keskittää kehittäjien huomiota pelien vähemmän esillä oleviin osiin. Myös oman osaamisen kehittäminen ja pelituntumaa parantavien mekaniikoiden suunnittelemisen ja toteutus ovat opinnäytetyön pääta-voitteina.

## 2 Pelituntuman määrittely

Steve Swink määrittelee kirjassaan ”A Game Designer’s Guide To Virtual Sensation” pelituntuman koostuvan kolmesta eri osasta: reaaliaikainen kontrolli, spatiaalinen simulaatio ja viimeistely. Ilman niitä todellista pelituntumaa ei voi Swinkin mukaan saavuttaa ja täten todellisen pelituntuman olemassaolo on Swinkin mielestä rajoitettu tietynkaltaisiin peleihin (1 s. 9).

Pelituntuman määritelmä ja teoria tässä opinnäytetyössä perustuvat pitkälti tähän kirjaan.

### 2.1 Reaaliaikainen kontrolli

Reaaliaikainen kontrolli tarkoittaa jonkinlaisen peliobjektin, yleensä pelihahmon, jatkuvaa ohjaamista. Pelatessa pelaaja ikään kuin sulautuu pelihahmon kanssa yhteen; pelihahmosta tulee pelaajan oma representaatio pelin sisällä. Pelaaja ei erota pelihahmoa omana entiteettinään, vaan siitä tulee suora pelaajan jatke. Jotta hahmon ohjaaminen voitaisiin luokitella jatkuvaksi, voidaan jatkuva kontrolli määritellä käyttämällä ihmisen vastesykliä. (1 s. 35)

Kun ihminen havaitsee jotain, kestää tiedolla noin 100 millisekuntia, että tieto voidaan käsitellä. Tiedon käsittelyyn kuluu noin 70 millisekuntia, jonka jälkeen kuluu taas 70 millisekuntia motorisiin toimintoihin. Nämä sykliet toimivat riippumattomina toisistaan, eli voimme havainnoida uutta tietoa jo ennen kuin motorinen toimintamme on täysin loppunut. (1 s. 36) Kuva 1 havainnollistaa ihmisen havaintojaksoa.



Kuva 1. Vuorovaikutteisuuden kaavio.

Esimerkkinä voidaan käyttää pelaajaan ja tietokoneen välistä vuorovaikutusta. Pelaaja havainnoi peliruudun näyttämää kuvaa, josta ärsyke saapuu aivoille käsiteltäväksi. Tämän jälkeen pelaajan motorinen toiminta liikuttaa kättä painamaan ohjaimen nappia. Napin painallus johtaa tietokoneen antaman ärsyksen muuttumiseen ja sykli alkaa alusta. Jotta peli voi reagoida pelaajan syötteeseen tarpeeksi usein näyttääkseen sulavalta liikkeeltä, täytyy siis pelin virkistystaajuuden olla vähintään kymmenen kehystä sekunnissa. Korkeampi virkistystaajuus johtaa peliobjektien sulavampaan liikkeeseen ruudulla.

Jotta pelihahmon ohjaaminen tuntuisi mahdollisimman sulavalta, täytyy pelin reagoida pelaajan syötteeseen vähintään joka 100 millisekunnin välein (1 s. 36). Jos tässä kestäisi pidempään, pelaaja joutuisi odottamaan oman syötteensä vaikutusta pelissä, uutta ärsykettä. Kun uusi ärsyke saapuu tämän aikaraamin sisällä, ei pelaaja ehdi havainnoimaan syötettään yksittäisinä toimintoina, vaan ne muodostavat sulavan jatkumon, reaaliaikaisen kontrollin. Mikäli peli ei vastaa pelaajan syötteeseen jatkuvasti alle 100 millisekunnin viiveellä, ei pelaajalla ole samanlaista kontrollia. Pelaajan fuusioituminen pelihahmon kanssa särkyy, pelihahmon ohjaaminen tuntuu verkaiselta ja pahimmillaan pelihahmon toiminnat voivat tuntua kokonaan irrallisilta pelaajan nappulan painalluksista. Pelissä voi kuitenkin esiintyä pidempi tauko, jonka aikana pelaaja ei voi vaikuttaa peliin (1 s. 47). Esimerkiksi pelihahmon syöksyssä pelissä tiettyyn suuntaan pelaaja voi menettää suoran kontrollin hetkeksi, kunnes pelihahmo on lopettanut syöksyn. Tärkeintä on, että pelaaja tietää, minkä takia tämä tapahtui, ja että se on suora seuraus hänen toiminnastaan. Pelaaja on itse päättänyt, että pelihahmo syöksyy. Reaaliaikaisuus voi siis rikkoontua ilman, että pelaaja tuntee menettävänsä kontrollia.

Tämän lisäksi pelaajan syötteen tulisi olla mahdollisimman johdonmukaista. Pelaajalle tulisi olla aina selkeää, mitä jokainen syöte aiheuttaa pelin sisällä. Mikäli saman nappulan painaminen aiheuttaa aina eri vasteen pelihahmon ollessa samassa tilassa, ei pelaajalla ole oikeaa kontrollia pelihahmon suhteen. Pelihahmo reagoi pelaajan syötteeseen, mutta pelaaja ei tiedä, mikä syötteen tulos on etukäteen.

Tällaista määritelmää seuraten voidaan osoittaa, onko pelissä reaaliaikaista kontrollia vai ei. Esimerkiksi vuoropohjaiset pelit eivät yleensä saavuta reaaliaikaista kontrollia, sillä niissä interaktion tiheys on paljon hitaampi. Interaktio loppuu näissä peleissä usein ilman pelaajan aikomusta, kun pelin viholliset toimivat omalla vuorollaan ja pelaaja joutuu odottamaan omaa vuoroaan ennen kuin voi taas toimia pelin kanssa vuorovaikutuksessa. Tämän lisäksi pelihahmojen ohjaaminen tapahtuu usein erillisen käyttöliittymän kautta, jolloin pelaaja ei ole enää suorassa vuorovaikutuksessa pelihahmon kanssa. Pelaaja ei ohjaa hahmoa, vaan antaa sille käskyn tehdä jotain.



## 2.2 Spatiaalinen simulaatio

Spatiaalinen simulaatio tarkoittaa pelin sisällä simuloitua tilaa ja sen kanssa vuorovaikuttamista. Simuloidun tilan tulee olla kirjaimellinen tila, jossa pelihahmo voi liikkua vapaasti (1 s. 4). Simulaatio on abstrakti, jos pelihahmon liikkuminen on rajoitettu esimerkiksi ruudukolla liikkumiseen, kuten vaikkapa shakissa. Simulaatio ei voi saavuttaa hyvää pelituntumaa abstraktilla simulaatiolla, sillä pelaajan liikkumista on rajoitettu tavalla, joka on peliympäristöstä riippumaton, arbitraarinen.

Simulaatiolla tarkoitetaan sitä, että pelihahmo pystyy vuorovaikuttamaan tilan kanssa. Yksinkertaisimmassa muodossaan se voi tarkoittaa, että pelihahmo törmää tasossa oleviin seiniin ja lattiaan. Näitä elementtejä hyödyntäen pelaaja kykenee päättämään pelihahmon näennäisen nopeuden ja koon. Samalla tavalla kuin pelihahmon kyvyt pelin sisällä määrittelevät hahmoa, määrittelee peliympäristö myös hahmoa (1 s. 4). Hahmo ei voi hypätä seiniä pitkin ilman seinien olemassaoloa, eikä liukua mäkiä alas ilman mäkiä. Parhaimmillaan pelaaja pystyy jättämään peliympäristöön oman jälkensä, joka on havaittavissa kauan aikaa kyseisen tapahtuman jälkeen. Pelaaja ymmärtää, että tila on fyysinen, permanentti ja että se tunnistaa pelihahmon olemassaolon (2).

Pelaajan täytyy myös aktiivisesti itse havaita tilaa (1 s. 5). Jos pelaaja ei ole hallussa omasta havaitsemisestaan, esimerkiksi pelihahmon tai pelikameran liikuttamisesta, hyvää pelituntumaa ei voida saavuttaa, sillä tilaa ei varsinaisesti simuloida. Peli vain toistaa valmiiksi määritellyt tapahtumat lineaarisessa järjestyksessä. Peli voi simuloida tilaa ja sen sisällä olevat peliobjektit voivat vaikuttaa toisiinsa ja reagoida ympäristöönsä monimutkaisesti, mutta pelaaja ei koe sitä suoraan itse.

## 2.3 Viimeistely

Viimeistelyä ovat kaikki efektit, jotka eivät suoranaisesti muuta pelimekaniikkoja tai vaikuta pelin pelaamiseen mekaanisella tasolla (1 s. 5; 3). Tämä tarkoittaa yksinkertaisuudessaan kaikkea grafiikkaa, ääniefektejä, musiikkia jne. Sen sijaan viimeistely vaikuttaa siihen, miten pelaaja havaitsee ruudulla tapahtuvaa toimintaa. Kuva 2 havainnollistaa viimeistelyn vaikutusta pelimaailmaan.



Kuva 2. OCDAn maailma ilman ylimääräistä viimeistelyä verrattuna viimeistelyyn maailmaan.

Ilman viimeistelyä pelaajan käsitys pelihahmosta olisi vähemmän vakuuttava. (1 s. 6). Pelaaja ohjaisi abstraktia peliohjainta ja pelaajan täytyisi oppia pelihahmon ohjaamisen säännöt täysin alusta alkaen. Kun tähän objektiin aletaan lisäämään viimeistelyä, aletaan saamaan eheämpi kuva hahmosta. Jos lisätään kuva, joka kuvaa humanoidia, voidaan olettaa pelihahmon olevan ihminen. Kun kuvia vaihdetaan nopeaan tahtiin, nähdään, että pelihahmo liikkuu vaikkapa juoksemalla. Askeliin voidaan vielä liittää pelihahmon juostessa ääniefekti, joka vaikuttaisi pelihahmon askelten painon tuntuun (4). Viimeistelyefektien avulla pelaaja voi tehdä oletuksen siitä, minkälaiselta pelihahmon ohjaaminen tulee tuntumaan ennen kuin on koskenutkaan peliin. Jatkamalla viimeistelyn päällekkäisyyttä saadaan yhä selkeämpi kuva tästä objektista ja miten se käyttäytyy ja reagoi sen ympärillä olevaan maailmaan kanssa. Nämä efektit eivät ole tarpeellisia pelin toimimiselle, mutta ne ovat kriittisiä sille, miten pelaaja havainnoi pelin toimimista ja minkälaisen tunteen hän saa pelistä.

Kun pelihahmo juoksee, viimeistely määrittää, millä tavalla hahmo juoksee. Jos hahmon animaatio on hidas ja kömpelö, saadaan vastaavanlainen kuvan hahmosta. Jos taas animaatio on nopeaa ja sulavaa, hahmo vaikuttaa paljon ketterämmältä. Samalla tavalla ääniefektit kuvaavat hahmon liikettä. Keveät sipsuttelevat ääniefektit antavat erilaisen kuvan kuin järeät ja lujat tömähdykset. Viimeistelyn lisääminen voi mullistaa pelihahmon tunteen täysin muuttamatta pelin toimintaa millään tavalla.

### 3 Pelituntuman metriikat

Vaikka pelituntumaa ei voidakaan suoranaisesti mitata, voidaan tarkastella, mitkä eri asiat vaikuttavat edellä mainittuihin pelituntuman kategorioihin. Jos pelituntumaa ei jaeta mitattaviin osiin, täytyisi hyvä pelituntuma rakentaa alusta alkaen jokaisessa peliprojektissa, tai hyödyntää aikaisemmassa projektissa saavutettua pelituntumaa.

Pelituntuman metriikoita hyödyksi käyttäen voimme tarkastella olemassa olevien pelien pelituntumaa ja oppia, miten eri osat niissä avustavat hyvän pelituntuman luomista (1 s. 81). Tarkastelun jälkeen voidaan opittua hyödyntää saman metriikan kautta toiseen projektiin ilman, että pelituntuman rakentaminen alkaisi täysin nollasta.

#### 3.1 Syöte ja vaste

Syöte tarkoittaa pelaajan syöttämiä komentoja pelille sekä pelaajan käyttämää fyysistä ohjainta, miten nappulat vastustavat pelaajan sormen painallusta ja minkälaisen tuntuman ne antavat pelaajalle. Ohjaimen paino ja sen rakentamiseen käytetyt materiaalit vaikuttavat myös sen tuntuun. Pelaaminen perinteisellä konsoliohjaimella tuntuu täysin erilaiselta kuin saman pelin pelaaminen puhelimen kosketusnäytöllä johtuen edellä mainituista muuttujista. (4; 5)

Pelille annettua syötettä voidaan mitata ja tallentaa, mikä tekee syötteestä hyvin konkreettisen metriikan. Tallennettua syötettä ja videokuvaa pelin pelaamisesta voidaan tarkastella samaan aikaan, jolloin voidaan helposti nähdä, mitä nappuloita pelaaja painoi milläkin hetkellä (6). Tarkastellessa voidaan havaita tietynkaltainen syöte, joka voi olla vaikkapa hyvin hankala pelaajalle tuottaa luotettavasti.

Erilaiset ohjaimet antavat erilaisia mahdollisuuksia syötteelle. Tavallinen ohjaimen nappula sisältää vain kaksi tilaa, pohjaan painettu ja ylhäällä oleva. Ohjaussauva taas sisältää miljoonia mahdollisia syötteitä, kun sen horisontaalista ja vertikaalia sijaintia käytetään yhdessä. Ohjaussauvan molempien akselien arvo vaihtelee sulavasti arvojen 1 ja -1 välillä. Kaikkien painikkeiden yhdistelmä muodostaa ohjaimen mahdollisien syötteiden syvyyden (1 s. 87). Hyvää pelituntumaa on hankala saavuttaa, mikäli ohjain ei mahdollista tarpeeksi syvällistä syötettä pelille. Ohjain, jossa on vain yksi nappula, vaikeuttaisi ensimmäisen persoonan räiskintäpelin pelaamista. Kuva 3 kuvastaa mahdollista pelaajan syötelaitetta.



Kuva 3. Donkey Kong Jungle Beat -pelin mukana tuleva peliohjain muuttaa kyseisen pelin tuntu-  
maa huomasti pakottamalla pelaajan lyömään bongorumpuja ohjatakseen pelihahmoa. (7)

Muuttamalla nappuloita, joita painamalla pelihahmo suorittaa verbejä, syötteen tuntuma vaihtuu. Erilaiset näppäinyhdistelmät samalla ohjaimella voivat tuntua hyvinkin erilaiselta. Sen sijaan, että jokaisella pelihahmon verbille olisi oma nappulansa ohjaimessa, voidaan kaikki verbit toteuttaa samalla nappulalla, mutta vaatia, että pelaaja painaa samaan aikaan eri suuntiin ohjaussauvalla suorittaakseen eri verbejä. Tällaisia näppäinyhdistelmiä suunnitellessa tulisi ottaa huomioon ihmisille luontaiset käden sijainnit. Jos jokin pelihahmon verbi vaatii kolmen eri nappulan painamista samaan aikaan, nappulat voidaan hajauttaa molemmille käsille, sen sijaan, että pelaajan tulisi painaa saman käden sormilla kaikkia nappuloita. Jos pelihahmon verbi vaatii monen nappulan painamista nopeasti putkeen, nappuloiden tulisi olla peliohjaimessa paikoissa ja järjestyksessä, joka on fyysisesti miellyttävä suorittaa. (5)

Vaste määrittelee, miten peli prosessoi pelaajan syötettä ja näyttää sen ruudulla. Peli voi muuntaa hyvinkin yksinkertaisen syötteen, kuten nappulan pohjaan painalluksen, monimutkaiseksi liikkeeksi peliruudulla (1 s. 89). Jos peli ei prosessoisi syötettä ollenkaan, pelihahmon verbit olisivat hyvin rajoittuneita. Pelihahmon liikuttaminen nappulaa painamalla liikuttaisi pelihahmoa aina yhden koordinaatin verran yhden kehyksen aikana, sillä nappulan painaminen lähettää pelille arvon yksi. Mikäli peli prosessoi syötettä, voi pelihahmo nappulaa painaessa hitaasti kiihdyttää tiettyyn vauhtiin ja lopulta saavuttaa ja pitää maksiminopeutensa. Nappulasta irti päästettäessä pelihahmo voisi hitaasti liukua, kunnes se lopulta pysähtyy. Tällä tavoin peli muuttaa yksinkertaisesta

syötteestä monimutkaisen vasteen pelaajalle. Pelaajan syötteen syvyydestä ja vasteen monimutkaisuudesta saamme kuvan pelin herkkyydestä syötteelle (1 s. 135).

Voidaan myös tarkastella, millaisen vasteen syöte saa aikaan. Syöte voi vaikuttaa pelihahmoon suoraan painalluksen tapahtuessa tai se voi muuttaa jotain globaalia muuttujaa pelin sisällä, kuten painovoimaa, niin kauan kuin nappulaa pidetään pohjassa. Samoin syöte voi vaikkapa kääntää jonkin muuttujan pääläelleen pelin sisällä painalluksen hetkellä, mutta kääntää sen takaisin, kun nappulasta päästetään irti.

Vaste ei myöskään ole staattinen asia, vaan sitäkin voidaan muuttaa pelin sisällä dynaamisesti. Esimerkiksi pelihahmo, joka kykenee juoksemaan ja hyppimään voisi moduloida vastettaan, mikäli se astuisi pelimaailmassa toiselle planeetalle, jossa painovoima ja ilmasto olisivat erilaisia. Pelihahmo tekee edelleen täysin samoja asioita, mutta vasteen muuttaminen kontekstin perusteella muuttaa pelihahmon tuntumaa (1 s. 133). Kuva 4 havainnollistaa, miten konteksti voi muuttaa vastetta.



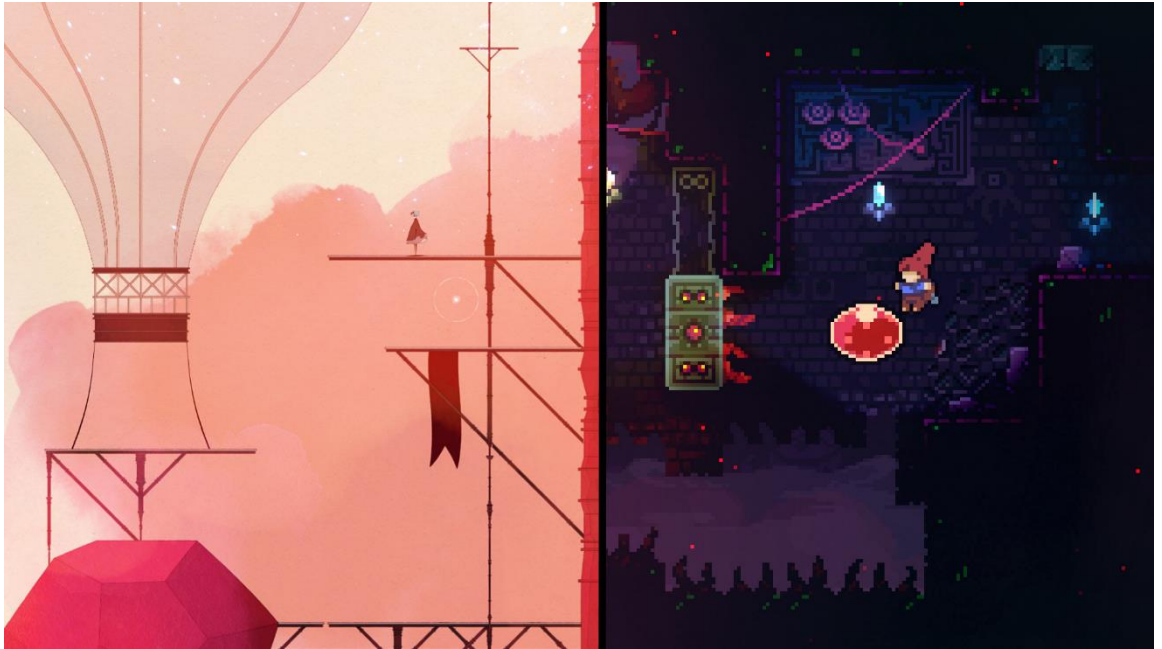
Kuva 4. Pelissä Chucksaw pelaajan syötteen vaste muuttuu täysin sahan ollessa tarpeeksi lähellä pelihahmoa. Pelaaja voi samalla painikkeella sekä napata sahan että heittää sen.

### 3.2 Konteksti

Konteksti käsittää simuloitun tilan ja sen vaikutuksen pelihahmoon ja pelaajaan. Peliympäristön koko suhteessa pelihahmon nopeuteen ja ketteryyteen muodostavat tilan ylimalkaisen tunnun,

kun taas tapa, jolla pelihahmo voi olla vuorovaikutuksessa tilan kanssa muokkaa tätä tunnetta syvällisemmin. (1 s. 92) Mikäli pelihahmo ei reagoisi tilaan millään tavalla, tilasta tulisi kuitenkin suurimmaksi osaksi merkityksetön pelaajalle. Tilan viimeistely voisi edelleen vahvistaa jotain tiettyä tunnetta, mutta pelihahmon kyvyt olisivat merkityksettömiä, sillä tila ei vaadi niiden hyödyntämistä. Ei ole väliä, jos pelihahmo voi hypätä, jos ympäröivä tila on täysin tasainen. Pelihahmon kyvyille ei ole kontekstia. Kun peliympäristö pakottaa pelihahmoa hyödyntämään kykyjään, tilan konteksti syntyy (8).

Mikäli peliympäristö on hyvin laaja, jossa pelihahmo voi liikkua vapaasti ilman suurempia esteitä, tila tuntuu rennolta ja pelaajan on helpompi tutkia aluetta. Pelihahmon liikenopeudesta riippuen tilan tuntu voi vaihdella vielä tästä. Jos pelihahmo on hidas, mutta ketterä, tilassa liikkuminen on edelleen helppoa, mutta paikasta toiseen liikkumisessa kestää enemmän aikaa. Tila tuntuu suuremmalta ja pelaajan huomio siirtyy pelihahmosta sitä ympäröivään tilaan pelihahmon liikuessa esteestä toiselle. Jos pelihahmo kiitäisi suurella nopeudella lentäen tilan läpi, tilan tuntu ei olisi välttämättä enää yhtä laaja, sillä pelihahmolla ei kestä yhtä kauan sen läpi kulkemisessa. Samoin tilan avoimet esteet saattavat muuttuvat yhä helpommiksi, mutta pelihahmo pääsee esteestä toiselle myös nopeammin. Koska pelaaja kohtaa haasteita useammin, tilan tuntu muuttuu vaarallisemmaksi. Jos pelihahmon nopeus pidettäisiin samana, mutta tilasta tehtäisiin ahtaampi, pelihahmo saattaisi alkaa tuntumaan kömpelöltä. Suuren nopeudensa takia pelihahmo törmäilisi seiniin liikuessaan ympäri ahdasta tilaa. Kuva 5 visualisoi kuinka peliympäristön skaala vaikuttaa tilan tuntumaan.



Kuva 5. Pelihahmon koko verrattuna kuvattuun pelimaailman skaalaan vaikuttaa tilan tuntuun. Kuvat peleistä GRIS ja Celeste.

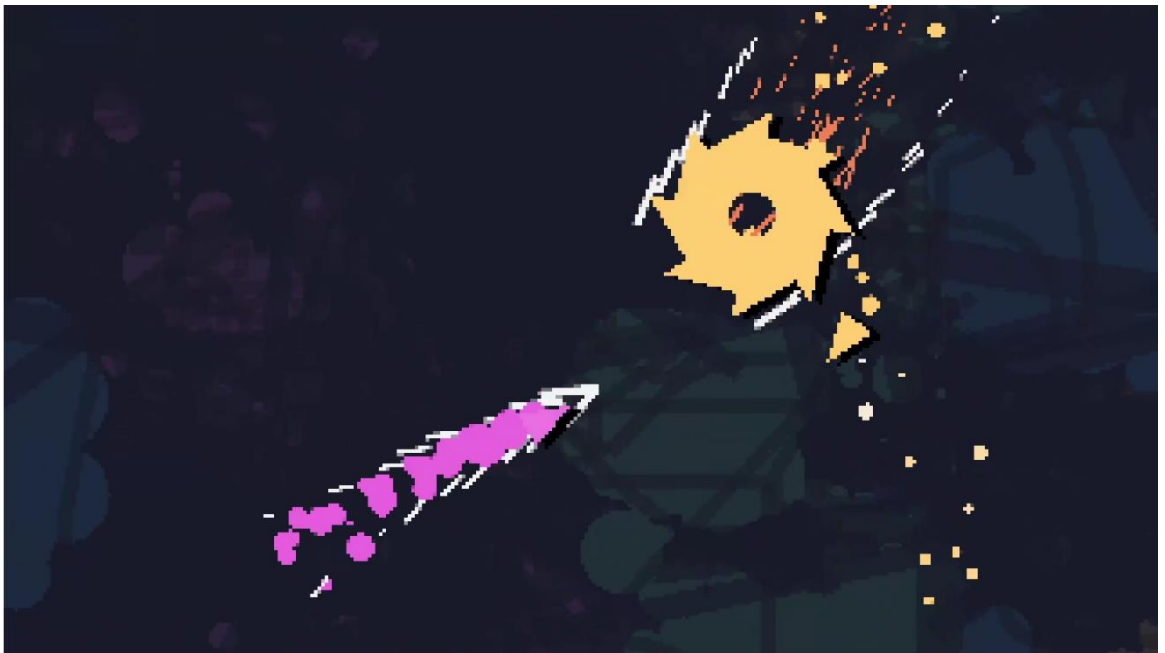
Pelaajan interaktion laatu tilan kanssa vaikuttaa myös tilan tuntumaan (1 s. 148). Tasohyppelypelissä jos pelihahmo kuolee seinään osuessaan, tila tuntuu hyvin vaaralliselta. Mikäli pelihahmo yksinkertaisesti pysähtyy, kun se koskee seinää, sama tila muuttuu vähemmän vaaralliseksi. Jos epäonnistumisella on pelaajalle suuria seurauksia, helpotkin haasteet muuttavat tilaa paljon vaarallisemmaksi. Vaarallisen ja turvallisen tilan määrän suhde tasossa määrittelee, miltä se tuntuu. Jos pitkässä tasossa on vain pari turvallista paikkaa, jossa pelaaja voi huohtaa ja saada tarkistuspisteen edistymiselleen ja joihin pääsee vain vaarallista tilaa ylittämällä, tila tuntuu uhkaavalta pelaajalle. Turvallisen tilan määrää lisäämällä pelaaja voi rentoutua ja liikkua tilassa vaapammin. Tila tuntuu kutsuvammalta.

### 3.3 Metafora ja viimeistely

Metafora määrittelee, miten asioita esitetään pelissä. Jos esine pelissä näyttää ulkopuolisesti viivulta, sen pitäisi pelaajan näkökulmasta myös toimia vipuna, eli pelihahmon pitäisi pystyä kääntämään sitä jollain tavalla. Pelaaja havaitsee peliobjektin sen vasteen, kontekstin ja viimeistelyn avulla ja tuo mukanaan omat odotuksensa esineen toiminnallisuudesta, joka perustuu pelaajan aikaisempiin kokemuksiin. Pelaajan odotukset eivät siis aina täsmää todellisuutta, vaan heidän omaa ymmärrystään peliobjektin toiminnasta, mikäli peliobjekti on oikeaan maailmaan perustuva. (1 s. 97; 9) Yhdessä ne muodostavat mielikuvan peliobjektista, jonka pitäisi täsmätä sen

toiminnallisuutta. Mikäli metafora ei vastaa peliobjektin toiminnallisuutta pelissä, rikkoo se kokonaiskuvan esineestä (10). Pelaajan odotus objektin toiminnallisuudesta ei olekaan enää yhtä lähellä sen oikeaa käyttäytymistä ja peliobjektin tuntuma kärsii.

Pelin graafinen ulkoasu vaikuttaa hyvin paljon pelaajan odotuksiin pelistä. Mitä realistisemmalta peli näyttää, sitä enemmän pelaaja odottaa esineiden käyttäytyvän kuten oikeassa elämässä. Tällöin vasteen ja kontekstin täytyy myös vastata oikeaa elämää hyvin tarkasti, jotta saataisiin oikeanlainen tuntuma. Jos taas pelin grafiikat ovat enemmän abstrakteja, pelaajalla ei ole yhtä tarkkaa odotusta pelin toiminnallisuudesta. Esineen muotokieli ja väri määrittävät peliobjektin ylimalkaisen tunteen ja mahdollisen suhteen pelaajaan, mutta tarkempia havaintoja on hankala tehdä (11). Tällöin vasteen, kontekstin ja viimeistelyn ei tarvitsekaan vastata esineen toiminnallisuutta yhtä tarkasti, jotta eheä metafora säilyisi. Samalla tavalla, jos peligrafiikka esittää olemassa olevia asioita, mutta hyvin tyylitellysti, eivät pelaajan odotukset niiden toiminnallisuudesta ole yhtä tarkat. (1 s. 171) Auton liikkuminen oikeassa maailmassa on hyvin erilaista kuin auton käyttäytymisen piirretyssä. Kuvassa 6 saha Chucksawista.



Kuva 6. Saha on kuvattu hyvin yksinkertaisena ilman ylimääräisiä yksityiskohtia jättäen sen oletetun toimintatavan epämääräiseksi ensisilmäyksellä. Sahan representaatio on ikoninen, ei realistinen.

Viimeistely vahvistaa pelaajan käsitystä pelielementtien fyysisestä luonteesta. Viimeistelyefektin eri osat sallivat pelaajan muodostaa erilaisia havaintoja peliobjektista (1 s. 151; 12). Esine, joka värisee paikallaan, vaikuttaa siltä, että se voisi lennähtää muualle hetkenä minä hyvänsä, kun taas



täysin paikallaan oleva esine vaikuttaa passiiviselta. Jos metafora kertoo pelaajalle, mitä peliohjekti voisi tehdä, viimeistely kertoo, miten peliohjekti tekee kyseisen asian.

Viimeistely tulee usein ilmi pelaajalle viimeistään peliohjektin ollessa vuorovaikutuksessa toisen peliohjektin kanssa. Esimerkiksi pelihahmon koskiessa peliohjektiä se katoaa. Viimeistelyefektien avulla määritellään, mitä peliohjektille implikoidaan tapahtuvan. Lennättämällä verta ympäristöön ja soittamalla huudahdusäänen on pelaajalle selvää, että koskettu peliohjekti kuoli. Jos taas soitetaan mystinen ääniefekti ja ohjekti katoaa hiljalleen nauraen jättäen usvaa jälkeensä, on ohjektin katoamisella täysin eri merkitys. Kuvassa 7 esimerkki Chucksawin pelihahmon viimeistelyefektistä.



Kuva 7. Pelihahmon syöksyessä sahaan törmäyksestä lentelee kipinöitä osoittaen pelihahmon sekä sahan olevan tehty kovasta materiaalista.

Kun kaikki viimeistelyefektit, grafiikka ja ääni, yhdistetään samaan esineeseen, pelaaja muodostaa näistä efekteistä yhtenäisen kuvan esineestä. Tätä mielikuvaa käyttäen pelaaja voi yrittää päätellä vaikkapa esineen mahdollista painoa ja tekstuuria sen ollessa vuorovaikutuksessa ympäristönsä kanssa. (1 s. 155) Vaikka viimeistelyefektit ovat tärkeitä pelihahmon verbien tuntumaan, on tärkeää, että efektejä lisättäessä kaikki efektit vahvistavat samanlaista metaforaa (10; 12; 13). Viimeistely ei kuitenkaan pyri antamaan tarkkoja, mitattavia ominaisuuksia peliohjektista, vaan ylimalkaisia tunteita niiden attribuuteista. Varsinkin kahden peliohjektin ollessa vuorovaikutuksessa, viimeistelyn suurin vaikutelma tulee peliohjektien erilaisista viimeistelyefekteistä, sillä pelaaja saa vertauspisteen viimeistelystä. Pelaaja ei voi esimerkiksi sanoa ohjektien tarkkaa painoa, mutta osaa sanoa kumpi on painavampi.

### 3.4 Säännöt

Pelin säännöt ovat käytännössä vain abstrakteja muuttujia, joilla on suhteita toisiin muuttujiin. Yhdessä ne muodostavat pelin säännöt. Yksinkertainen esimerkki sääntöjen vaikutuksesta pelituntumaan on vihollisten tekemän vahingon määrä pelihahmoon. Jos vihollinen tekee pelaajaan esimerkiksi yhden elämapisteen verran vahinkoa ja pelaajalla on sata elämapistettä, on vihollinen verrattain vaaraton pelaajalle. Jos tätä muuttujaa, vahingon määrää, nostetaan kymmeneen, vihollisen tuntuma pelin sisällä muuttuu täysin. Vihollisesta on tullut paljon vaarallisempi pelaajalle. Samalla kaikki peliympäristöt, jotka sisältävät tätä vihollistyyppiä, tuntuvat vaarallisemmilta.

Sääntöjä voidaan ajatella pelimekaniikkoina. Mitä enemmän yhteyksiä muuttujilla on toisiinsa, sitä monimutkaisempia pelin mekaniikoista ja niistä muodostuvista systeemeistä muodostuu. Jos edellä mainittuun esimerkkiin lisätään mekaniikka, jolla pelihahmo voi saada takaisin menetettyjä elämäpisteitä, tämä muuttaa jälleen kerran vihollisen tuntua pelaajalle. Mitä korkeammalla tasolla mekaniikka toimii, sitä laajemmin se vaikuttaa pelin tuntumaan. Elämäpisteiden takaisin saamisen mekaniikka vaikuttaisi kaikkeen pelissä tapahtuvan taistelun tuntumaan. (1 s. 180)

Samalla tavalla pelissä oleva kolikko voi yksinään olla hyödytön. Se ei vaikuta mitenkään pelin muihin muuttujiin. Jos kolikot kytketään johonkin muuhun muuttujaan, joka hyödyttää pelaajaa, tämä kytkös antaa kolikoille arvoa. Yhtäkkiä pelaaja haluaakin kerätä kolikoita. Kun kolikoille on annettu merkitys pelissä, se muuttaa monen muun asian tuntumaa. Tietyt tasot pelissä voivat muuttua paljon vaikeammiksi, jos pelaaja haluaa kerätä kaikki tasossa olevat kolikot, mutta kolikot ovat paikoissa, joihin pelihahmon on vaikea päästä.

Jos kolikoihin kytketty muuttuja on resurssi, joka antaa pelaajan edetä pelissä eteenpäin, muodostavat pelin säännöt yhdessä puitteet pitkäjänteiselle motivaatiolle (1 s. 99). Pelaajan tavoitteena on pelata peli läpi. Jotta pelaaja voi pelata pelin läpi, täytyy hänen edetä pelissä, jota varten hänen tarvitsee saada resurssia, joka mahdollistaa sen. Jotta pelaaja saisi näitä resursseja, täytyy hänen kerätä kolikoita. Muutaman muuttujan ja niiden välisten yhteyksien avulla voidaan siis luoda pelaajalle jatkuva motivaatio pelaamiseen, parantaen pelituntumaa saamalla pelaajan toiminnot tuntumaan tekemisen arvoisilta. Pelaajan lyhyen aikavälin motivaatio yhdistyy saumattomasti pitkäaikaiseen motivaatioon. Jos pitkäaikainen motivaatio on pelaajan sisäinen motivaatio, eikä pelin antama ulkoinen, arbitraarinen rajoite, kuten aikaraja, on se mielenkiintoisempi pelaajalle. Kun pelaajan tavoite on sidottu pelimekaniikkoihin, säännöt ovat mukana luomassa motivaatiota muillakin tavoilla kuin vain rajoittamassa pelaajan etenemistä pelissä (13).

### 3.5 Yhtenäisyys

Jotta peli saavuttaisi hyvän pelituntuman, on tärkeää, että kaikki edellä mainitut osat sopivat yhteen täydellisesti (1 s. 307). Pelkästään yksi pelin aspekti, joka erottuu joukosta voi pilata pelituntuman. Viimeistelyn ja vasteen tulisi vahvistaa koherenttia metaforaa pelihahmosta ja -maailmasta. Tämän lisäksi pelin kontekstin ja sääntöjen, eli ympäristön ja mekaniikkojen, tulisi luotsata pelaajaa pelaamaan kaikkein hauskimalla mahdollisella tavalla. Tämä tarkoittaa kaikkien pelihahmon kykyjen käyttämistä ja muiden pelimekaniikkojen hyödyntämistä pelatessa. Pelin tulisi olla suunniteltu pelaajan käyttämää ohjauslaitetta mielessä pitäen. Jos peliä ollaan tekemässä PC-alustalle, tulisi sen tuntua mahdollisimman hyvältä sekä ohjaimella että näppäimistöllä ja hiirellä pelattaessa.

Suuret ja näyttävät visuaaliset efektit eivät sopisi helposti peliin, jossa pelaaja tutkisi pientä saarta omalla rauhallisella tahdillaan. Mikäli pelin tarkoitus olisi vahvistaa tällaista rauhallisuuden tunnetta ja etsinnän halua, peli ei voisi myöskään olla kovin mekaanisesti vaikea pelata, etteivät mekaniikat estäisi pelaajaa tutkimasta ympäristöään. Vertaamalla pelituntuman metriikoita pelin toivottuun tunteeseen voidaan havaita ja ryhtyä korjaamaan pelin aspekteja, jotka eivät sovi yhtenäisesti samaan teemaan.

#### 4 Juissin määrittely

Juissilla tarkoitetaan mitä tahansa efektiä pelissä, joka antaa ylimääräistä palautetta pelaajan syötteeseen, jonkin peliobjektin tilan muuttumiseen tai uuden peliobjektin syntymiseen (13). Juissi on yleensä havaittavissa pelissä lyhyinä efekteinä, jotka katoavat yhtä nopeasti kuin ilmestyivätkin, kestäen usein vain muutamia kehyksiä, sen sijaan että olisivat jatkuvasti havaittavissa.

Juissi on pelin tapa varmistaa, että pelaaja tietää pelin vastaanottaneen pelaajan syötteen tai jonkin asian tapahtuneen pelimaailmassa (4). Se tekee itse syötteen antamisesta mahdollisimman tyydyttävää pelaajalle ja on parhaimmillaan, kun se yhdistää monta erilaista viimeistelyefektejä tuottamaan yhtenäisen kuvan pelaajan syötteestä tai sen tuloksesta (3). Juissi ei kuitenkaan rajoitu pelkästään viimeistelyefekteihin, vaan voi myös vaikuttaa varsinaisiin pelimekaniikkoihin. Esimerkiksi pelin pysäyttäminen muutaman kehyksen ajaksi lyönnin tapahtuessa voi antaa pelaajalle ylimääräistä miettimisaikaa taistelun aikana (2; 14). Tällaista efektiä ei yleisesti mielletä pelimekaniikkana, sillä sen vaikutus pelin pelaamiseen on vähäinen, joten se voidaan luokitella paremman puutteessa juissin alle. Pelin pysäyttämistä käytetään useimmiten pelissä tapahtuvan iskun osuman korostamiseksi. (12; 15) Laajempi kuva juissin kasaamisesta saadaan, kun tarkastellaan kuvaa 8.



Kuva 8. Esimerkki juissista pelissä Hollow Knight pelihahmon lyödessä vihollista. (14)

Kuvassa pelaajan syötteen seurauksena pelihahmo hyökkää. Hyökkäämisen merkiksi pelihahmolle on animoitu erillinen hyökkäysanimaatio. Kun pelihahmon hyökkäys osuu viholliseen, voimme alkaa luettelemaan eri efektejä.

Puhtaasti viimeistelyefekteinä toimivat:

- Miekan iskun mukainen pitkä ja kaareva viiva, joka vahvistaa miekan iskun liikerataa.
- Vihollisen kohdalle sijoittuvat viivat rastin muodossa, jotka liikkuvat vihollisesta ulospäin.
- Oranssi, räjähtävä pilvi, joka katoaa ilmaan.
- Oranssit pisarat, jotka lentävät vihollisesta ulos ja putoavat maahan.
- Oranssi, välähtävä hehku vihollisen ympärillä.
- Vihollinen välähtää oranssina.
- Vihollisen animaatio lennättää vihollisen ruumista taaksepäin ilmassa.
- Miekan lyönnin ääniefekti.
- Miekan viholliseen osumisen ääniefekti.
- Vihollisen kuolemisen ääniefekti.

Tämän lisäksi muutama efekti vaikuttaa hieman myös itse peliin, mutta eivät ole mekaniikoita:

- Vihollinen ripottelee valuuttaa ilmaan.
- Peli pysähtyy muutaman kehyksen ajaksi.
- Kamera tärisee.

Peli yhdistää siis yhteen pelaajan syötteeseen, napin painallukseen, jopa 13 erilaista efektiä, jotka yhdessä luovat mahdollisimman tyydyttävän kuvan pelihahmon lyönnistä vihollista kohti. Juisiefektit ovat ikään kuin hyvin dramatisoituja, äärimmäisiä versioita asioista, jotka voisivat tapahtua oikeassa elämässä (9). Kaiken tämän tavoitteena on saada pelaajalle mahdollisimman selkeä audiovisuaalinen palaute hänen syötteestään ja sen vaikutuksesta pelimaailmaan (12). Pelihahmo hyökkää syötteen suorana seurauksena, jonka pelaaja voi havaita pelihahmon hyökkäysanimaatiosta ja siihen liitetystä efekteistä. Tämän jälkeen pelaaja voi tuntea osuvansa viholliseen, kun myös viholliseen liitetään efektejä. Efektien perusteella pelaaja tietää myös, että vihollinen on kuollut, sillä se lensi ilmaan ja tiputti valuuttaa. Pelaaja saa siis vahvistetun palautteen peliltä, että syöte on vastaanotettu, sitten pelaaja saa palautetta siitä, osuiko pelihahmo iskullaan vai ei.

Viimeiseksi viimeistelyefektit kertovat pelihahmon iskun laadusta, sen vaikutuksesta pelimaailmaan, miten vihollinen reagoi iskuun.

Koska kyseinen esimerkkipeli ei pyri realistiseen grafiikkaan, nämä efektit sopivat peliin, vaikka ne ovatkin hyvin liioiteltuja. Efektit on tehty yhtenäisesti samanlaisella piirretyllä tyylillä kuin pelin muutkin toiminnalliset grafiikat. Juuri tämä tyyllittely vahvistaa pelin omaa metaforaa, hahmon ja vihollisten fyysistä luonnetta ja mahdollista käyttäytymistä.

Joissakin tapauksissa on hankala eritellä, onko jokin asia välttämättä juissia vai ei. Aikaisemmassa esimerkissä voitaisiin väittää vihollisesta putoavan valuutan lentäminen olevan pelimekaniikka, sillä pelihahmo joutuu tämän seurauksena keräämään valuutan suuremmalta alueelta vihollisen kuoltua.

Kun juissia tarkastellaan pelituntuman metriikoiden kautta, voidaan sanoa juissin olevan melkein eksklusiivisesti viimeistelyä. Tämä viimeistely vaikuttaa pelin metaforaan hyvin paljon. Ilman näitä efektejä ei voida sanoa mitään pelihahmon hyökkäyksen laadusta tai siitä, miten vihollinen reagoi hyökkäykseen. Ilman näitä efektejä ei olisi ollenkaan selvää, onko hahmo edes hyökkäämässä tai osuuko lyönti ollenkaan viholliseen. Pelaaja ei voi siis silloin tehdä oletusta syötteesä vaikutuksesta syötteen jälkeen. Jos edellä mainittu tilanne riisuttaisiin kaikesta viimeistelystä, lopputuloksena pelaaja voisi havainnoida vain kaksi abstraktia peliobjektia, joista toinen katoaa äkkinäisesti ilman mitään näennäistä interaktiota.

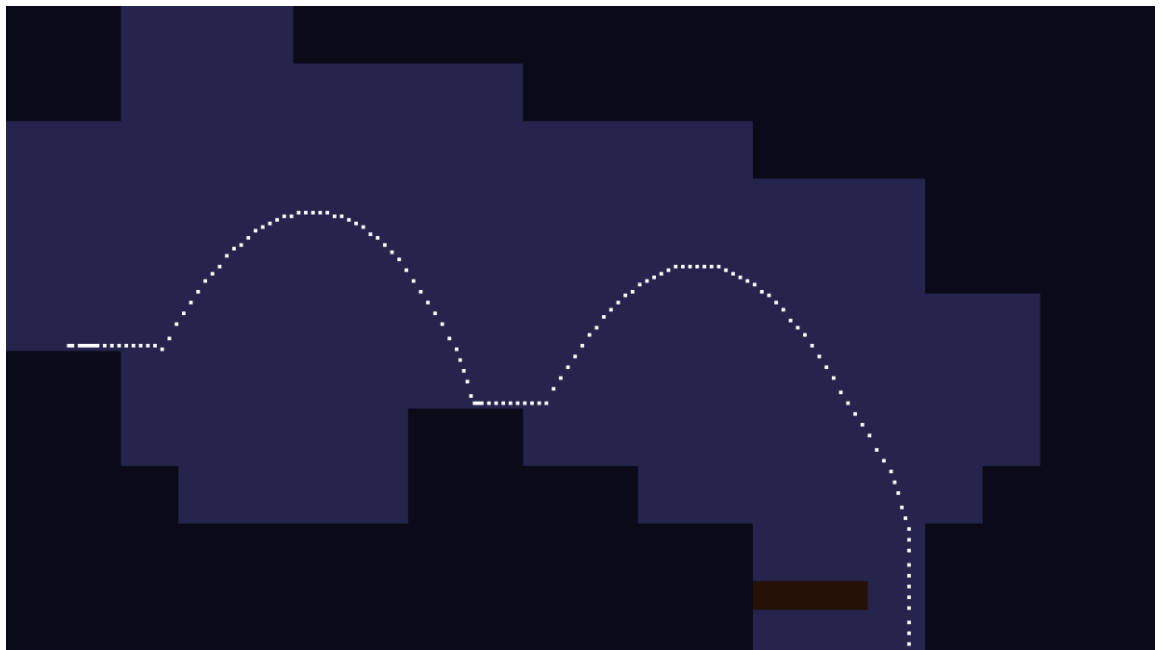
## 5 Öljyn määrittely

Öljyä käsitteenä voidaan tarkastella juissin kautta. Kun juissi keskittyy asioiden korostamiseen ja palautteen antamiseen, öljyn tehtävä on olla mahdollisimman huomaamaton pelaajalle (16). Öljyn tehtävä on minimoida pelaajan kokemaa vaivaa tai turhautuminen pelaajan pyrkiessä päämääräänsä pelissä (17). Öljyn implementointi vaihtelee erittäin paljon myös pelityypin mukaan. Toimintapeleissä öljyä käytetään usein pelaajan syötteiden yksinkertaistamiseen tai mekaniikkojen parempaan esittämiseen, kun taas strategiapeleissä voidaan öljyä käyttöliittymän virtausta.

Jotta peliä voidaan öljytä, täytyy ensin määritellä, mitä pelaajan täytyy osata kyseisessä pelissä, minkälaisia taitoja peli yrittää pelaajalle opettaa. Kun tämä on selvitetty, voidaan seuraavaksi kysyä, minkä osan pelistä tulisi olla haastava pelaajalle. Yleisimmin pelaajalle opetettava taito tai pelaajan tavoite on laajemmassa kontekstissa, kuten resurssien hallinta tai vihollisten väisteleminen ja tason loppuun pääsy. Jos pelissä on monta eri systeemiä, on tiedettävä missä systeemissä pelin syvin tarkoitus on. Jos pelissä on sekä tasohyppelyä että taistelua, toinen systeemi väistämättä tukee toista. Jos taistelu on vaikeaa pelissä ja pelaaja viettää eniten aikaa sitä tehdessä, tasohyppelyelementtien tulisi olla pelaajalle helpompaa, ikään kuin täytettä taistelujen välissä. (18)

Pelin päämekaniikkaa voidaan öljytä luontevammaksi ja mekaanisesti helpommaksi, jotta sen jatkuva tekeminen olisi miellyttävämpää pelaajalle. Öljyllä yritetään poistaa ylimääräiset askeleet pelaajan ja mekaniikan väliltä, jotta pelaaja voi hyödyntää ja olla vuorovaikutuksessa mekaniikan kanssa mahdollisimman vaivattomasti. Mekaniikan alkuperäinen haaste ja kompleksisuus yritetään kuitenkin pitää ennallaan, se vain esitetään pelaajalle mahdollisimman intuitiivisella tavalla. Räiskintäpelissä ammuntaa voitaisiin öljytä tekemällä kriittisistä osumista helpompia suorittaa. Peli voi näkymättömästi nykäistä pelihahmon ampumia luoteja vihollisten päitä kohti. Tuloksena pelaaja voi ampua vihollisen kriittisestä pisteestä hiukan ohi, mutta peli korjaa tämän virheen muuttamalla luodin lentorataa hieman. Toimintapelissä peli voi automaattisesti kääntää pelihahmon hyökkäyksen lähellä olevaa vihollista kohti (19). Taktisessa pelissä öljyäminen voisi tarkoittaa yksiköiden siirtämiseen vaadittujen syötteiden määrän minimoimista. Pelaaja saa haluamansa vasteen tapahtumaan pienemmällä määrällä vaivaa ja pelin tahti soljuu nopeammin.

Päämekaniikkaa tukevia mekaniikkoja voidaan öljytää rankemmin. Koska kyseessä on tukeva elementti, pelaaja tulee yleensä kohtaamaan mekaniikkaa taukona päämekaniikasta. Tämän mekaniikan kanssa vuorovaikuttamisen tulisi siis olla pelaajalle rentouttavaa vaihtelua. Tässä kontekstissa öljyäminen voi tarkoittaa mekaniikan suoraa virtaviivaistamista, yksinkertaistamista. Ylimääräinen kompleksisuus poistetaan yksinkertaisemman suorittamisen tieltä. Esimerkiksi kolmannen persoonan räiskintäpeleissä räiskintää tuetaan usein tasohyppelyllä. Tasohyppelystä riisutaan hyvin paljon oleellisia mekaniikkoja, kuten jopa itse hyppely. Manuaalisen hyppäämisen sijaan pelihahmo hyppää automaattisesti ennalta määrättyyn kohteeseen, kun pelihahmo lähestyy maan tai lattian reunaa. Tasohyppely on vain keino päästä toiseen paikkaan, jossa suorittaa räiskintää tai tapa, jolla tutkia ympäristöä. Kuva 9 osoittaa yhtä keinoa öljytää tasohyppelyä.



Kuva 9. Syötepuskurin hyödyntäminen hypätessä tason reunalta.

Vaikka pelihahmo on jo juossut pois tason päältä, pystyy hän hyppäämään silti. Tämä ei ole millään tavalla realistista, mutta se tekee pelaajan elämästä paljon helpompaa antamalla pelaajalle pienen virhemarginaalin hypyn syötettä antaessa. Jos pelaaja painaisi hyppäysnappulaa muutama kehyksen verran liian myöhään, eikä pystyisi hyppäämään, se tuntuisi pelaajasta hyvin epärealilta. Pelaajan silmissä hän teki kaiken täydellisesti, sillä hän ei pysty havaitsemaan näin pientä ajallista eroa. (20) Pelaajan tarkoitus oli ilmiselvästi hypätä. Ellei pelin tarkoitus ole tehdä jokaisesta hypystä mahdollisimman haastavia, pelituntuma kärsii tällaisista hetkistä. Pelaajaa voidaan kuitenkin auttaa sallimalla hänen hypätä hetken aikaa, kun hän on jo alkanut putoamaan tasolta. Tekemällä pieni puskuri, jonka sisällä pelaajan sallitaan epäonnistua, saavutetaan parempi peli-



tuntuma (21; 19). Haasteen ei tulisi olla itse syötteen tekemisessä ja hyppypainikkeen painamisessa, vaan hahmon paikoittamisessa oikein hyppyä varten ja hahmon hallitseminen ilmassa hypyn jälkeen. Pelaaja ajattelee hahmon ohjaamista, ei syötteen antamista. (5)

Öllyämisen tavoitteena on siis minimoida kitka pelaajan syötteen, tavoitteen ja pelin antaman vasteen välillä ilman pelin haasteen poistamista. Pelin öljyminen tekee pelistä anteeksiantavamman pelaajan virheitä kohtaan, eikä vaadi jatkuvaa täydellisyyttä.

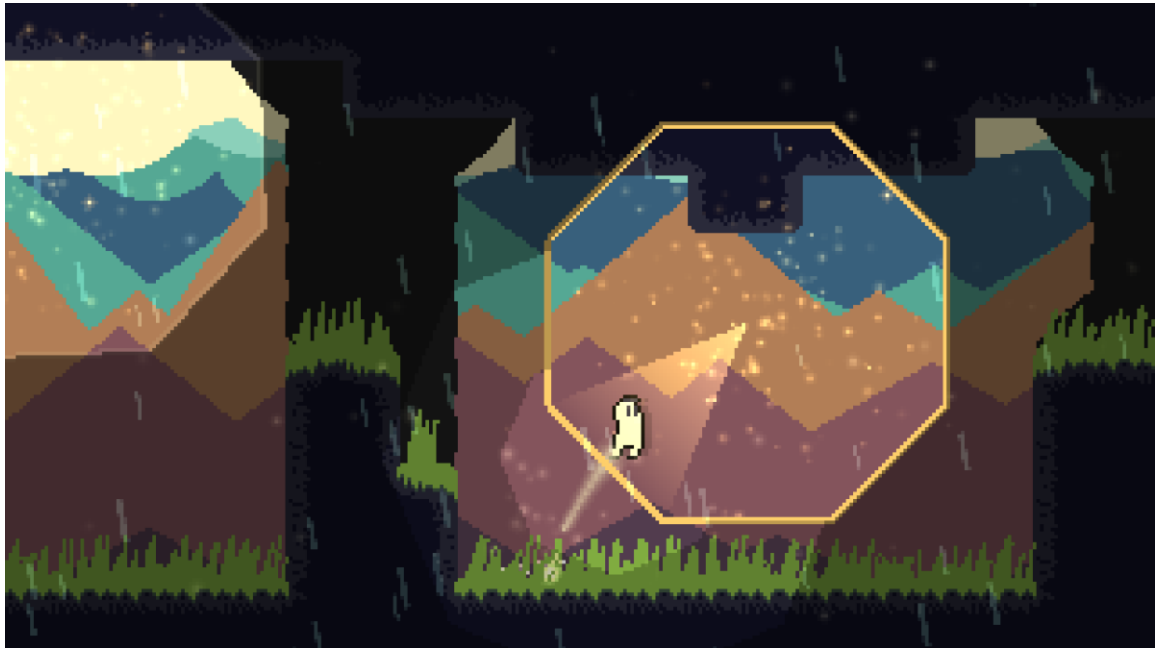
Kun peli on hyvin öljytty, se tunnistaa pelaajan syötteen mahdollisimman luotettavasti, pelihahmon ohjaamisesta tulee johdonmukaista ja tarkoituksellista. Pelin antama vaste pelaajan syötteleelle ei välttämättä täsmää itse syötteeseen, mutta täsmää pelaajan syötteen tarkoitettuun tulokseen. Peli yrittää ikään kuin arvata jokaisen pelaajan syötteen halutun lopputuloksen ja auttaa saavuttamaan se, sen sijaan, että se noudattelisi syötettä täydellisesti (19; 22). Öllyminen voidaan siis toteuttaa muokkaamalla vastetta.

Tällainen pelin öljyminen väistämättä vaikuttaa myös pelin kontekstiin. Jos pelaajan liikkumista helpotetaan öljymällä, ympäröivä tila muuttuu helpommaksi, mikä vaikuttaa sen tuntuun pelaajan näkökulmasta. Hypyt, jotka olivat ennen öljymistä hyvin tarkkoja ja anteeksiantamattomia ovatkin muuttuneet hypyiksi, jotka pelaaja voi toteuttaa suurella varmuudella lopputuloksesta. Mikäli pelin tavoite ei ole tehdä jokaista pelaajan tekemää hyppyä vaikeaksi, pelituntuma parane.

Pelin säännöt muuttuvat öljymisen tuloksena myös vaihtelevammiksi. Yhtäkkiä pelin yleinen sääntö tai mekaniikka, ”Pelaajan tulee seisoa maassa hypätäkseen”, ei pädekään jokaisessa tilanteessa. Öllyämisen tuloksena pelihahmoa koskevat säännöt alkavat suosimaan pelaajaa sen sijaan, että olisivat täysin staattisia. (20)

## 6 Tasohyppelypele

OCDAn tasohyppelypele, jossa pelihahmo pystyy hyppäämisen ja juoksemisen lisäksi käyttämään hyödykseen tasoihin sijoitettuja syöksyalueita. Alueiden sisällä pelihahmo voi syöksyä joko alueen keskipistettä kohti tai siitä poispäin. Pelin tavoitteena on yksinkertaisesti edetä tasosta toiseen, keräten matkalla tasoihin piilotettuja esineitä. Kuvassa 10 kuvakaappaus OCDAsta.



Kuva 10. OCDAn pelihahmo hyppäämässä syöksyalueen sisällä.

Peli on tehty haastavaksi, mutta silti hyvin anteeksiantavaksi pelaajalle. Peli keskittyy pelihahmon tarkkaan sijoittautumiseen syöksyalueiden sisällä ja syöksyn oikeaan ajoittamiseen. Tarkistuspiteitä on sijoitettu tasoihin usein ja pelaajan epäonnistuessa uudelleenyritys on nopeaa.

Peliä tehdessä keskityttiin pelihahmon ohjauksen tuntuman hiomiseen, sekä pelihahmon vasteen kautta että tasosuunnittelua tehdessä. Pelaajan syötettä yritettiin pitää mahdollisimman yksinkertaisena pelihahmon verbien määrään verrattuna pelaamisen helpottamiseksi (23). Syötteen vastetta yritettiin kehittää tuntumaan pelaajalle luotettavammaksi ja mekaanisesti helpommaksi toteuttaa.

## 6.1 Syötepuskurin hyödyntäminen

Syötteen puskurointi tarkoittaa sitä, että peli ei välittömästi unohda syötettä sen antamisen jälkeen. Sen sijaan syöte tallennetaan myöhempää käyttöä varten puskuriiin. Täten pelaaja voi antaa pelille syötteen hetkenä, jona pelihahmo ei kykene antamaan vastetta syötteelle, mutta heti kun pelihahmo kykenee reagoimaan syötteeeseen uudestaan, se voi etsiä uutta syötettä puskurista. (20) Mikäli puskurista löytyy syöte, jonka pelihahmo pystyy toteuttamaan sen nykyisessä tilassa, toteutetaan syötteen mukainen vaste ja puskurit tyhjennetään.

Esimerkkinä voidaan käyttää pelihahmoa, joka kykenee hyökkäämään vaikkapa lyömällä. Pelaaja antaa syötteen pelille eli painaa painiketta, joka korrespondoi hyökkäämiseen ja pelihahmo aloittaa lyöntianimaation. Pelihahmon lyöntianimaatio kestää jonkin aikaa ja mikäli pelaaja antaa uuden syötteen lyödäkseen animaation aikana, pelihahmo ei voi reagoida siihen, koska se on jo valmiiksi lyömässä. Kun lyöntianimaatio loppuu, pelihahmo ei hyökkää uudelleen, koska peli ei tiedä pelaajan aikeesta hyökätä uudelleen, sillä syöte on jo unohdettu.

Vaikka pelaaja haluaa hyökätä uudelleen, peli ei toteuta pelaajan toivetta, koska hän antoi syötteen hieman etuajassa. Tämä virhe pelaajan syötteenä on usein niin pieni, että pelaaja ei pysty havaitsemaan sitä ja pelaaja tulkitsee tapahtuman siten, että hän antoi pelille syötteen oikein, mutta peli ei vain osannut tulkinnut sitä (20). Vaikka pelaaja onkin väärässä, hän tuntee olonsa petetyksi. Tulevaisuudessa pelaaja saattaa turvautua hyökkäyspainikkeen jatkuvaan rämpyttämiseen, jotta peli tulkitsisi hänen syötteensä oikein.

Mikäli syöte säilytetään puskurissa hetken aikaa, pelaajan ei tarvitse olla täydellinen syötteenä kanssa, eikä myöskään rämpyttää painiketta. Kun hyökkäysanimaatio on päätynyt, peli tarkistaa puskurin ja toteuttaa seuraavan hyökkäyksen heti, kun se on pelihahmolle mahdollista. Pelaajan syötteen tarkoitus ja pelin vaste täsmäävät.

Puskuri ei voi kuitenkaan säilyttää syötettä ikuisesti. Puskurin kokoa täytyy rajoittaa ja hyödyllinen tapa saavuttaa tämä on käyttää rengaspuskuria. Puskurille määritetään koko ja haluttu syöte tarkistetaan joka kehyksen aikana ja tallennetaan puskuriiin. Kun päästään puskurin loppuun, siirytään takaisin puskurin alkuun ja uusi syöte kirjoitetaan vanhan datan päälle. Kun puskuria tarkistetaan syötteen varalta ja syöte löytyy, puskurit tyhjennetään datasta. Syötteen tallentaminen jatkuu tämän jälkeen normaalisti. Puskurit voi säilyttää kaikenlaista pelihahmolle relevanttia syötettä. Useimmiten puskuriiin tallennettaisiin kaikki syöte, joka aktivoi pelihahmon eri verbejä. Kuva 11 havainnollistaa rengaspuskuria.

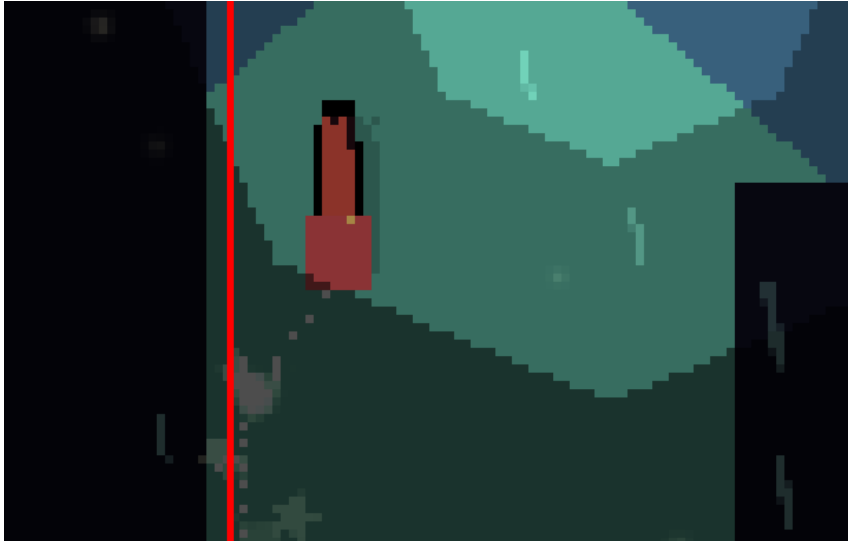


Kuva 11. Visualisaatio rengaspuskurista, jossa voidaan säilyttää yhdeksän arvoa samanaikaisesti.

OCDAssa pelihahmon verbit ovat hyvin rajoitettuja. Pelihahmo voi ainoastaan suorittaa erilaisia hyppyjä tai syökyä. Tämän lisäksi melkein kaikki pelihahmon verbit ovat kontekstista riippuvaisia, joten pelihahmolla ei usein ole montaa vaihtoehtoa verbien käytön suhteen. Pelaajaan mahdollinen syöte ei ole kovin monimutkainen ja tuloksena pelissä käytetty syötepuskuri on hyvin yksinkertainen. Ainoastaan hypyn ja syöksen painikkeiden syöte puskuroidaan.

Pelihahmo pystyy hyppäämään, kun se seisoo maan päällä tai on ilmassa seinää vasten. Syötteen puskuroida hypyn kontekstissa OCDAssa tarkoittaa sitä, että kun pelaaja on laskeutumassa aikaisemmasta hypystä tai pudotuksesta, pelaaja voi painaa hyppypainiketta ilmassa hieman ennen kuin pelihahmo osuu maahan. Kun pelihahmo lopulta laskeutuu, se hyppää uudestaan välittömästi. Mikäli pelaaja on painanut hyppypainiketta suurin piirtein laskeutumisen aikaan, voidaan tulkita pelaajan haluavan pelihahmon hyppäävän heti laskeutumisen jälkeen (21).

Sama logiikka pätee seinähyppyihin. Maahan osuminen korvataan vain seinään osumisella. Tämän lisäksi pelaajan ei kuitenkaan tarvitse koskaan ikinä fyysisesti koskea seinää. Riittää, että pelihahmo pääsee tietyn etäisyyden päähän seinästä, jolloin pelihahmo voi jo hypätä siitä. Koska OCDAssa pelihahmo hyppii usein seinästä toiseen edestakaisin, pelaaja saattaa jo ennen seinähyppyä yrittää liikkua vastakkaiseen suuntaan, jonka takia pelihahmo ei usein koske seinään olleensa hyppyjen välissä. Koska pelaajan tarkoitus on kuitenkin jälleen kerran hypätä, voidaan pelaajan seinähypyn vaatimuksia helpottaa tällä tavalla. Kuvassa 12 visualisoidaan tämä etäisyys.



Kuva 12. Etäisyys seinästä, jonka sisällä pelihahmo voi silti hypätä seinästä.

Lopuksi puskuroidaan pelihahmon syöksy. Sen puskurointi toimii lähestulkoon samalla tavalla kuin hyppyjenkin. Pelaaja voi painaa syöksypainiketta ennen kuin pelihahmo on oikeasti tarpeeksi lähellä syöksyalueutta. Kun pelihahmo saavuttaa alueen, se syöksyy heti alueen osoittamaan suuntaan. Pelihahmo tuntuu täten ohjautuvan paremmin, koska sen käyttäytyminen, eli vaste, vastaa pelaajan aikeita useammin.

Syötteen puskuroinnin lisäksi voidaan puskuroida pelin sisällä muita muuttujia, jotka rajoittavat pelaajan verbejä. OCDAn tapauksessa puskuroidaan pelaajan tilaa. Tämän efekti pelin pelaamiseen on käänteinen syötepuskurista. Pelaaja voi antaa syötteen liian myöhään ja puskurista tarkistetaan, olisiko pelihahmo voinut suorittaa syötettä vastaavan verbin puskurin ajalta. (20) Tällainen puskurointi on kuitenkin paljon tilannekohtaisempaa, sillä kaikkea syötettä ei voida käsitellä rekursiivisesti. Voi olla, että pelihahmon nykyinen tila ohittaa syötteen tärkeydessä, eikä syötettä siis huomioidakaan.

Esimerkkinä voidaan käyttää tilannetta, jossa vihollinen osuu pelihahmoon hyökkäyksellään. Pelihahmo ottaa vahinkoa ja siirtyy erilliseen tilaan, josta sen täytyy toipua, ennen kuin se voi tehdä mitään. Vaikka pelaaja painaisikin painiketta, jolla pelihahmo hyökkäisi heti tämän vahingon ottamisen jälkeen, pelihahmo ei voi siirtyä tästä tilasta toiseen syötteen takia. Vahingon ottamisen tila on tärkeydeltään suurempi kuin mikään muu tila, joten siitä ei voida siirtyä pois syötettä puskuroimalla.

OCDAssa pelihahmon hyppyä on puskuroitu tällä tavalla. Jos pelihahmo putoaa tasolta hypäämättä, voi pelihahmo kuitenkin vielä hypätä hetken aikaa. Pelaajat yrittävät usein hypätä juuri tasojen reunalta, mutta luontaisen virhemarginaalin vuoksi joskus pelaajan syöte on hieman myöhässä ja pelihahmo ehtii pudota tasolta. Puskurointi korjaa tämän ongelman, sillä puskurin aikana

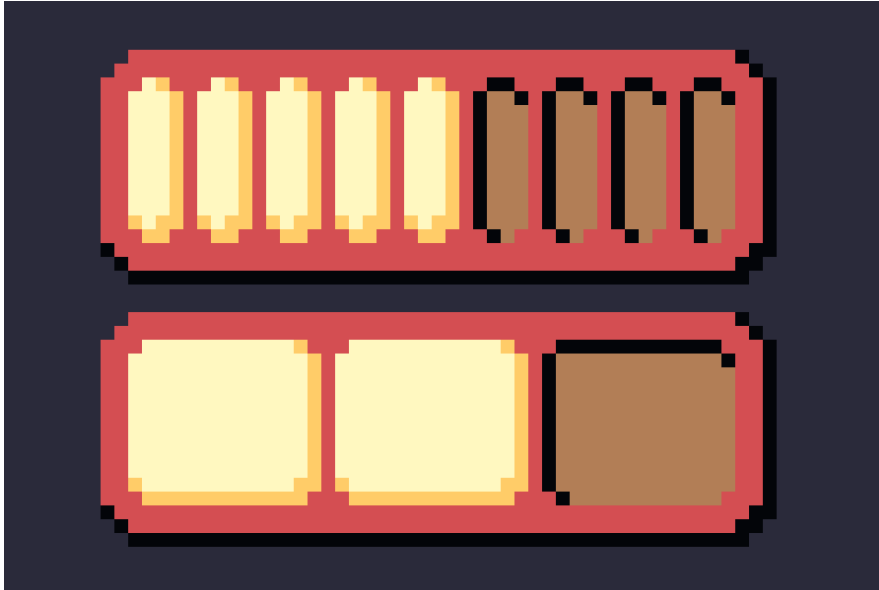
pelaaja voi vielä hypätä ja puskuri on tarpeeksi lyhyt, että pelaaja ei välttämättä huomaa sitä. Jos puskuri on liian pitkä, pelaaja voi huomata sen ja saattaa kohdella sitä ikään kuin uutena tason reunana (21). Silloin ongelma toistuu, kun pelaaja yrittää saada kaiken irti puskurista ja painaa taas hyppypainiketta liian myöhään. Tällaista puskurointia kutsutaan yleisesti kojoottiajaksi, mutta termiä voidaan käyttää myös muunlaiseen rekursiiviseen puskurointiin. Seinähyyt ja syöksyalueet hyödyntävät samankaltaista rekursiivista puskurointia. Puskuroinnin ansiosta pelaaja voi edelleen hypätä hetken aikaa poistuttuaan seinän vierestä. Pelihahmo voi myös syöksyä hetken aikaa sen jälkeen, kun se on poistunut syöksyalueelta.

Kaiken tämän syötteen puskuroinnin tuloksena pelihahmon toiminnat tuntuvat mahdollisimman sujuvilta ja luotettavilta. Puskurointi laajentaa pelaajan aikaa antaa pelille syöte, jotta pelaajan ei tarvitse joka kerta oikeasti onnistua antamaan pelille syötettä täysin oikeaan aikaan. Syötettä voidaan puskuroida käytettäväksi myöhemmin, mutta myös pelinsisäisiä muuttujia voidaan puskuroida, jotta pelaajan syötteen vaste voidaan aktivoida retroaktiivisesti.

## 6.2 Tapahtumien asteittaisuus

Pelihahmon verbien tulisi olla helppoja ymmärtää pelaajalle. Samoin verbien tuloksien pitäisi joka tapauksessa olla ennustettavissa, kun pelaaja suorittaa näitä verbejä. (24) Mikäli verbeillä voi olla monta lopputulosta riippuen pelaajan onnistumisen laadusta, voivat verbien seuraukset olla pelaajalle ennalta-arvaamattomia, etenkin jos verbien suorittaminen on vaikeaa.

Monipuolisten verbien tuloksia voidaan tehdä ennustettavimmaksi tekemällä tuloksista asteittaisia. Jos pelihahmo pystyy vaikkapa lataamaan hyökkäystä ja hyökkäyksen voima on suoraan sidottu latauksen määrään, on hyökkäyksen tarkkaa voimaa hankala määrittää, vaikka se olisi visualisoitu pelaajalle. Latauksen määrä, ja seurauksena hyökkäyksen voima, vaihtuvat jatkuvasti. Jakamalla latausmäärän esimerkiksi kolmeen osaan pelaajalla on konkreettinen tavoite jokaisella hyökkäyksellä. Sen sijaan, että pelaaja ajattelisi tekevänsä epämääräisesti suuren, voimakkaan hyökkäyksen, pelaaja voi ajatella tekevänsä nimenomaan kolmannen tason hyökkäyksen. Asteiden määrää ja latauksen nopeutta säätämällä voidaan asettaa verbille haluttu tuntuma. Asteittäisessä latauksessa pelaajalla on myös enemmän aikaa ajoittaa lataaminen tietylle asteelle. Koska pelaaja pystyy luotettavasti suorittamaan saman tason hyökkäyksen ja täten tietämään pelihahmon verbin tuloksen, pelituntuma paranee. Pelaajan syötteellä on selkeästi erilaisia vasteita ja pelaajan on helpompi toteuttaa haluttu vaste. Kuvassa 13 näytetään käytännön esimerkki asteittäisyydestä.



Kuva 13. Pelaaja haluaa osua latauspalkin keskelle. Suurempi asteittaisuus moninkertaistaa pelaajan mahdollisuuden osua.

On kuitenkin tärkeä mainita, että verbin asteittaisuus ei ole aina objektiivisesti paras toteutustapa monimuotoiselle verbille. On monia tapoja saada verbien vaste luotettavammaksi ja usein verbin koko ekspressiivisyys voi olla rakennettu sen liukuvaan skaalaan. Tällaisessa tapauksessa asteittaisuus voisi rajoittaa tai yksinkertaistaa verbiä liikaa. Verbien asteittaiset tulokset ovat kuitenkin helppo tapa tehdä verbistä lähestyttävämpiä ja intuitiivisia, kun asteittaisuus on indikoitu hyvin.

OCDassa verbin asteittaisuus tulee ilmi pelihahmon syöksyssä. Sen sijaan, että pelihahmo syöksyisi aina alueen keskustaa kohti, syöksyn suunta on jaettu kahdeksaan osaan. Pelihahmon sijainti suhteessa syöksyalueen sijaintiin pyöristetään syöksyn syötteen tapahtuessa. Sen sijaan, että pelihahmolla olisi 360 eri suuntaa, johon syöksyä, sillä on kahdeksan. Koska yksittäisen sektorin koko on suurempi, pelihahmolla on enemmän pelivaraa suuntaa valitessa. Pelaaja tietää tismalleen suunnan, johon pelihahmo syöksyy nappia painaessaan. Vaikka pelaaja ei onnistuisikaan syöksyssä, hän kuitenkin tietää tismalleen mihin sektoriin hänen tulisi tähdätä, sen sijaan, että hänen täytyisi pätkäillä alueen sisällä tarkkaa sijaintia, johon tähdätä.

Koska pelimaailma muodostuu neliömäisistä lohkoista, kahdeksansuuntainen syöksy myös linjautuu yleensä paremmin kuin vapaasuuntainen syöksy. Jos pelaajan täytyisi syöksyä ohutta kuilua pitkin suoraan ylös, vapaasuuntaisella syöksyllä pelaajan virhemarginaali olisi paljon pienempi. Jo muutaman asteen deviaatio voisi koitua pelaajan kohtaloksi, kun taas asteittaisella syöksyllä pelihahmon täytyy vain osua yhdelle, virhemarginaalia leveämmälle sektorille, joka lähettää pelihahmon oikeaan suuntaan. Kuvassa 14 on taso OCDasta.



Kuva 14. Ilman syöksyn suunnan asteittaisuutta täsmälleen ylöspäin syöksyminen ahtaissa tiloissa olisi hankalaa.

Asteittaisuudesta on myös tärkeä viestiä pelaajalle viimeistelyn avulla, jotta eri asteet ovat mahdollisimman selkeät. Kun siirrytään asteesta toiselle, voidaan soittaa ääniefekti sen merkiksi. Lisäksi nykyistä astetta voidaan näyttää graafisesti, jotta pelaaja tietää aina, mikä aste on aktiivinen. Peli muistuttaa pelaajaa koko ajan vasteesta, joka tapahtuisi oikean syötteen tapahtuessa. OCDassa syöksyalueella oleva valaistu sektori indikoi pelaajalle, mihin suuntaan pelihahmo syöksyisi, kun syöksypainiketta painettaisiin. Indikaattori on aktiivinen jo ennen kuin pelihahmo on syöksyalueen sisällä, jotta pelaajalla on aikaa nähdä indikaattori ja säätää pelihahmon sijainti oikein syöksyä varten.

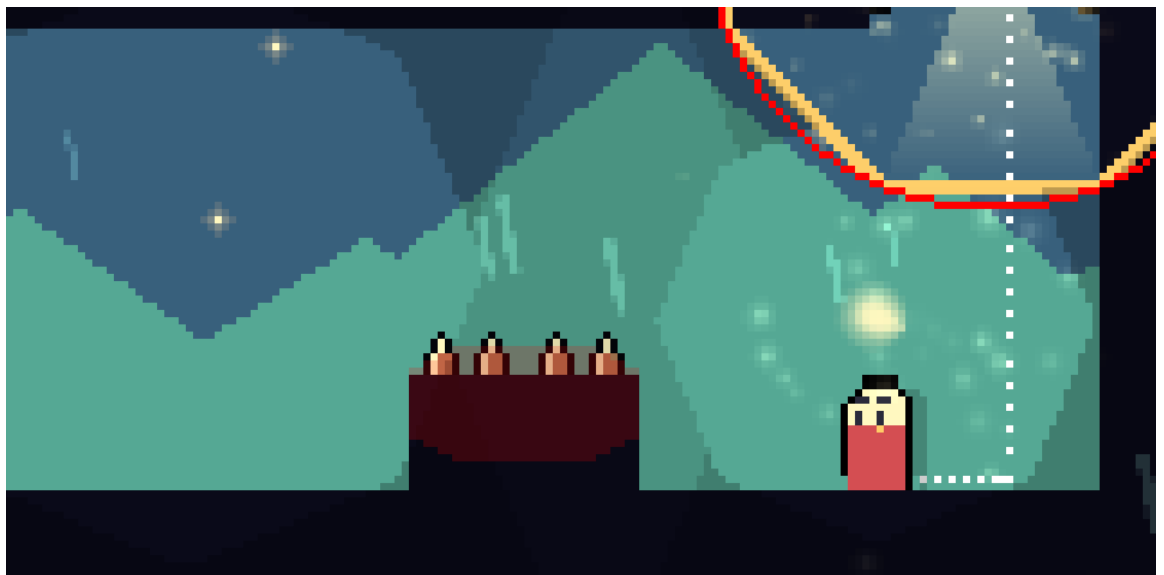
### 6.3 Törmäysalueiden suunnittelu

Peleissä grafiikat ja funktionaaliset elementit ovat usein toisistaan erillisiä. Peliobjekteille yleensä annetaan grafiikasta eroava törmäysalue. Kun kahden peliobjektin törmäysalueet kohtaavat, ne



voivat reagoida toisiinsa. Grafiikan ja törmäysalueiden kokojen suhde voi vaihdella hyvinkin paljon keskenään, ja tämä suhde vaikuttaa hyvin paljon peliohjainten tuntumaan. (20; 25) Olisi täysin loogista ajatella, että mitä paremmin peliohjainten grafiikka ja törmäysalue täsmäivät, sitä paremmin se toimisi pelin sisällä, koska ohjainten metafora olisi eheä. Tätäkin pelin aluetta voidaan kuitenkin suunnitella lempeämmäksi pelaajalle.

Pelaajalle haitallisten ohjainten, kuten vaikkapa ansojen tai vihollisten, törmäysalueita voidaan pienentää verrattuna niiden grafiikkaan. Esimerkkinä voidaan käyttää vihollisen hyökkäystä pelihahmoa kohtaan, joka on juuri hyökkäyksen kantaman rajalla, mutta osuu pelihahmoon. Pelaajan mielestä vihollisen hyökkäys ei ollut tarpeeksi lähellä osuakseen ja osuma tuntuu epärealistiselta, sillä pelaaja ei pysty toiminnan keskellä arvioimaan osumaa tarkasti. Mikäli vihollisen hyökkäyksen törmäysaluetta pienennetään hieman, tällaiset rajatapaukset suosivat pelaajaa ja pelaaja ei huomaa mitään erikoista. Pelihahmon voidaan antaa myös koskea hyökkäyksen törmäysalueeseen muutaman kehyksen ajan, jonka jälkeen pelihahmo vasta ottaa osuman (26). Vihollisen hyökkäyksen metafora ei ole välttämättä oikeasti yhtä eheä kokonaisuus, mutta pelaajasta se vaikuttaa eheältä tässä tilanteessa. Samanlainen efekti voidaan saavuttaa myös pienentämällä pelihahmon törmäysaluetta, johon vihollisten hyökkäykset yrittävät osua. Näitä törmäysalueita säätämällä viholliset voidaan saada tuntumaan reilummilta pelaajan silmissä. Kuvassa 15 näytetään OCDAn piikkiesteen toiminnallinen törmäysalue.

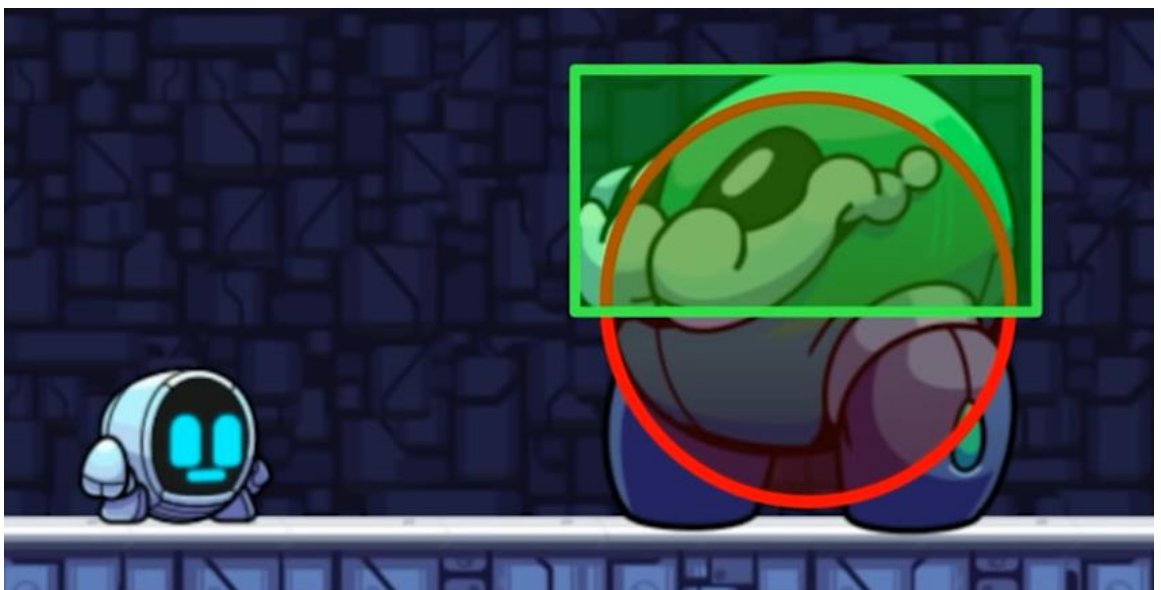


Kuva 15. OCDAn piikkiesteen toiminnallinen törmäysalue

Käänteisesti pelihahmon hyökkäysten törmäysalueita voidaan kasvattaa suhteessa grafiikkaan, jotta samanlaiset rajatapaukset pelihahmon hyökkäyksissä suosivat jälleen kerran pelaajaa. Jos pelaaja olisi juuri ja juuri iskenyt vihollisesta ohi, voi suurempi törmäysalue pelastaa tilanteen.

Kuitenkin törmäysalueiden koko täytyy pitää suurin piirtein grafiikkojen puitteissa. Mikäli törmäysalueet vaikuttavat täysin irrallisilta grafiikoista, peliohjelmien interaktio ei tunnu enää oikealta (27). Edellä mainitussa hyökkäyksessä pelaaja huomaisi grafiikoiden ja törmäysalueiden eron, ja illuusio törmäysalueiden ja grafiikoiden täsmämisestä menetettäisiin.

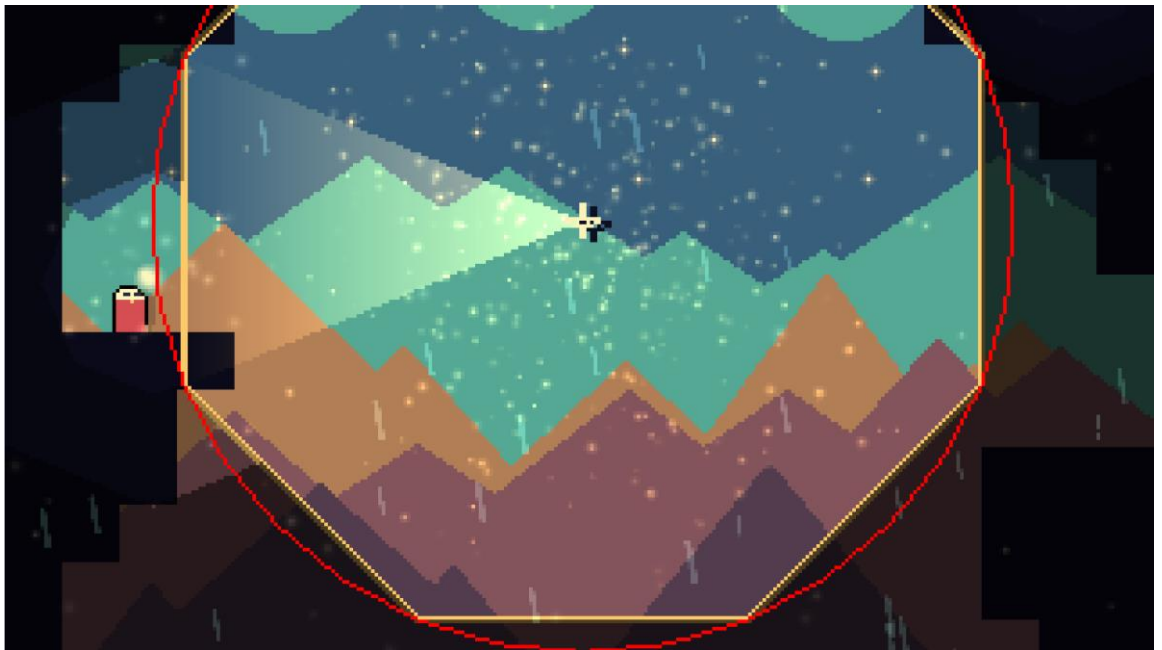
Mikäli pelihahmolla on useita tapoja vaikuttaa toiseen peliohjelmaan tai peliohjelma voi tehdä monta eri asiaa, eri toimintoille voidaan käyttää myös erillisiä törmäysalueita. Pelissä Levelhead viholliset vahingoittavat pelihahmoa koskiessaan sitä, mutta mikäli pelihahmo hyppää vihollisten päälle, se hyppää uudestaan ilmaan tuhoten vihollisen. Edellä mainittuja ohjeita noudattaen, vihollisen törmäysalue, jota käytetään pelihahmon vahingoittamiseen, on verrattaen pieni ja sijaitsee vihollisen alaosassa. Päälle hypätessä tarkistetaan kuitenkin ensin vihollisen päällä oleva törmäysalue. Se on muotoiltu leveämmäksi kuin vihollisen grafiikka ja istuu vihollisen päällä. Kun pelaaja yrittää hypätä vihollisen päälle, pelihahmo voi laskeutua hieman vihollisen sivulle ja silti onnistua hyppäämään sen päältä onnistuneesti. Koska vahingoittamiseen käytetty törmäysalue on ohuempi, pelihahmo voi hypätä helpommin vihollisen alta sen yläpuolella ja pudota takaisin sen päälle hypätäkseen uudelleen. Jos pelihahmo ja vihollinen olisivat molemmat paikoillaan, pelihahmo voisi yksinkertaisesti hypätä ylöspäin ja laskeutua automaattisesti vihollisen hypäykseen käytetyn törmäysalueen päälle ilman, että pelihahmo ottaisi vahinkoa. Ylimääräisen törmäysalueen tuloksena vihollisen metafora muuttuu; pelaaja näkee viholliset myös työkaluina tasohyppelyyn sen sijaan, että ne olisivat vain esteitä tasossa. Tällaisessa tapauksessa on tärkeää, että vihollisen viimeistely kommunikoi tällaisen interaktion olevan mahdollinen. Kuva 16 visualisoi monen törmäysalueen käyttöä vihollisessa.



Kuva 16. Pelin Levelhead erilliset törmäysalueet monipuolisille verbeille. (20)

Törmäysalueiden koon lisäksi voidaan muuttaa niiden muotoa. Pelimoottorista riippuen törmäysalueiden muotoa voidaan muuttaa suoraan, tai koota uudenmuotoinen alue monesta pienemmästä alueesta. Tällä tavalla voidaan muodostaa vaikkapa koveria törmäysalueita, mikäli niitä tarvitaan pelissä.

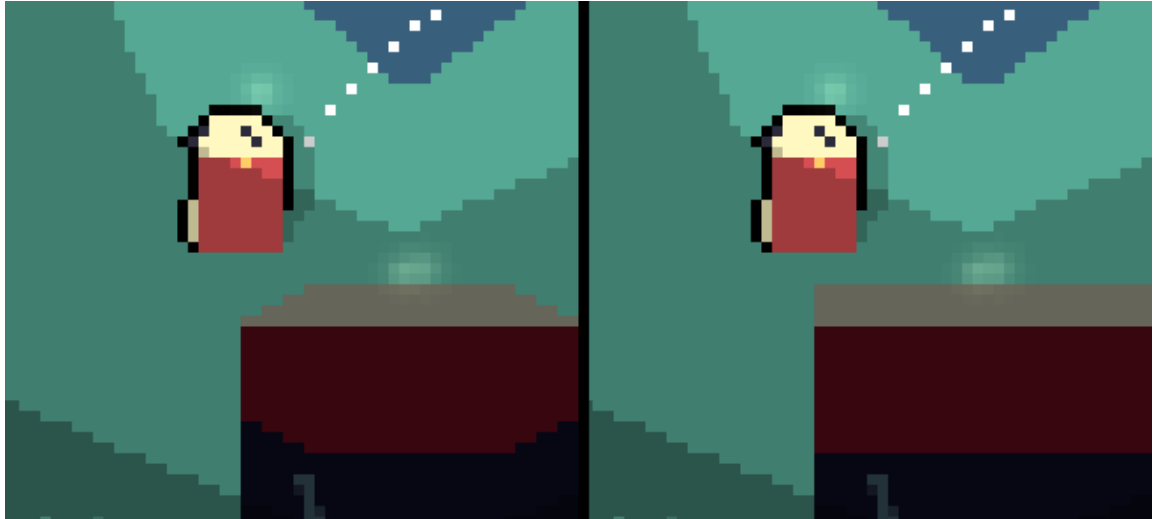
OCDAssa syöksyalueiden törmäysalue ei ole oktagonin muotoinen vaan ympyrä. Kuitenkin oktagonin kärjet kohtaavat ympyrän reunan. Käytännössä tämä tarkoittaa sitä, että törmäysalue ylittää oktagonin reunat ja tuloksena pelihahmo pystyy syöksymään hieman syöksyalueen ulkopuolella. Graafisesti pelaaja näyttää olevan juuri syöksyalueen rajalla ja pelaaja ei huomaa tätä pientä ylimääräistä tilaa alueiden ja pelaajan välillä. Pelaajasta tämä tuntuu luontaisesti reilulta. Pelaaja ei osaa erottaa grafiikan ja oikeiden törmäysalueiden pientä eroa pelatessaan, joten hän yrittää ajoittaa syöksynsä grafiikan mukaan. Suurempien törmäysalueiden ansiosta syöksyn toteuttaminen on helpompaa. Kuvassa 17 visualisoidaan tämä ylimääräinen tila.



Kuva 17. Syöksyalueiden grafiikat verrattuna törmäysalueeseen.

Tasohyppelypeleissä pelihahmot liikkuvat useimmiten sulavilla liikkeillä. Kun pelihahmo hyppää, hypyn vertikaali ja horisontaali nopeus yhdistettynä painovoiman kanssa muodostavat kaaren, ja koska hahmolla kestää hetki vaihtaa suuntaa ilmassa, samanlainen kaareva liike tapahtuu sekä pysty- että sivusuunnassa (21; 28). Koska pelihahmo liikkuu kaarissa, voidaan törmäysalueiden muotoa muuttaa sen mukaisesti, jotta ne mukailisivat tällaista liikettä. Kun pelihahmo hyppää esteen yli, suorakulmion muotoinen törmäysalue saattaa kohdata sen kulmista pelihahmon kanssa. Sen sijaan, että pelaajan täytyisi muuttaa tapansa liikkua pelin sisällä, on parempi, että

esteiden törmäysalueet myötäilevät pelihahmon luontevinta tapaa liikkua. OCDAssa pelihahmolle haitallisten esteiden törmäysalueet on muotoiltu ovaalin muotoisiksi myötäillen pelihahmon luontaista hyppykaarta. Esteiden metafora muuttuu jälleen kerran törmäysalueiden muokkaamisen seurauksensa vaarattommaksi. Kuva 18 näyttää eron ovaaleissa törmäysalueissa.

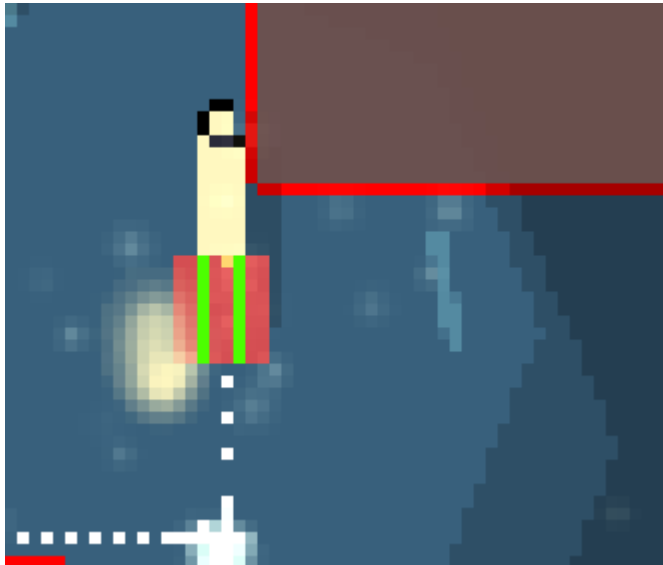


Kuva 18. Esimerkkitalanne, jossa pyöristetyt törmäysalueet pelastavat pelihahmon kuolemalta.

Peliympäristökään ei voi aina myötäillä pelihahmon liikkumiskykyä täydellisesti. Koska neliöistä muodostetun peliympäristön tekeminen on teknisellä tasolla helpompaa, se on usein valittu ratkaisu tasohyppelyä tehdessä. Se muodostaa kuitenkin ongelmia kaarimaisen pelihahmon liikkeen kanssa. Samalla tavalla kuin halutaan estää pelihahmon ja esteiden nurkkien törmäyslaatikoiden kohtaaminen, voidaan estää pelihahmon, lattian ja seinien törmäyslaatikoiden kohtaamista tietyissä tapauksissa.

Kun OCDAn pelihahmo osuu seinään, pelihahmon horisontaalinen liike pysähtyy. Samalla tavalla jos pelihahmo lyö päänsä kattoon, pelihahmon pystysuuntainen liike pysähtyy. Tätäkin sääntöä voidaan kuitenkin muokata hieman pelaajaa suosivaksi. Mikäli OCDAn pelihahmo lyö päänsä kattoon, se tarkistaa, onko törmäyskohdan lähellä vapaata tilaa. Mikäli pelihahmon törmäysalueen yläpuolella on paljon tyhjää tilaa ja sen törmäysalue osuu kattoon vain muutaman pikselin verran pelihahmon törmäysalueen reunoista, voidaan olettaa pelaajan halunneen hypätä seinää myötäillen ylös (20). Jotta pelaajan toive toteutuisi, voidaan pelihahmoa siirtää sivuun tämän muutamman pikselin verran, jotta se päätyy nätisti seinän viereen, jonka jälkeen pelaajan liikerata jatkuu normaalisti. Katon ja seinän kulma ikään kuin hiotaan pyöreäksi, jota pitkin pelihahmo liukuu seinää vasten (17). Koska ero pelihahmon sijainnissa verrattuna ennen siirtämistä on niin pieni, ei pelaaja huomaa pelihahmon siirtyneen automaattisesti sivulle. Mikäli pelihahmo on syöksymistilassa, voidaan tällaista virhemarginaalia kasvattaa vielä suuremmaksi. Koska tässä tilassa pelaaj-

jalla on vähemmän kontrollia pelihahmon liikkeestä ja pelihahmo liikkuu nopeasti, voidaan pelihahmoa siirrellä suurempia etäisyyksiä pelaajan puolesta, mikäli se osuu kattoon tässä tilassa. Kuva 19 näyttää esimerkin tilanteesta, jossa pelihahmoa siirretään.



Kuva 19. OCDAn pelihahmon käytännön vertikaali törmäysalue ympäristön kollisiossa.

Samanlaista kulman hiontaa voidaan soveltaa myös vaakasuunnassa. Jos pelaaja yrittää hypätä ylöspäin ympäristön reunuksen yli ja pelihahmon törmäysalueen alaosa osuu tähän reunukseen, mutta yltäisi muuten helposti tason päälle, voidaan pelihahmoa siirtää ylöspäin tarpeeksi tason päälle, jonka jälkeen sen liike jatkuu normaalisti. Hahmo pidetään liikkeessä ilman ylimääräisiä pysähdyksiä, jolloin liikkeen kaarimaisuus säilyy.

Käänteisesti jos pelihahmo on laskeutumassa hypystä ja on liikkumassa reunusta kohti, mutta ohittaa reunuksen pienellä määrällä, voidaan tämä havaita tarkistamalla laskeutuessa myös pelaajan vieressä oleva ympäristö. Mikäli havaitaan muutaman pikselin päässä oleva reuna, voidaan pelaajaa liikuttaa erotuksen verran, jotta se seisoo juuri ja juuri reunuksen päällä. Voidaan jälleen kerran olettaa pelaajan halunneen laskeutua reunukselle putoamisen sijaan (18). Tarkistus tulisi tehdä vain pelihahmon liikkumissuuntaa kohti, jotta pelaajaa ei liikutettaisi vaikkapa sen takana olevan reunuksen päälle vahingossa.

Jos pelihahmo ohittaa laskeutuessaan reunuksen suuremmalla erotuksella, OCDAn pelihahmo osaa myös kiivetä seinää ylös yhden neliön sivun verran. Vaikka teknisesti ottaen pelaaja on epäonnistunut hypyssään, peli antaa vielä yhden ylimääräisen oljenkorren johon pelaaja voi tarttua. OCDAssa jokaisessa hypyssä on yleensä monta osaa, ja jos pelaaja onnistuu muissa osissa, mutta ei pääse aivan tarpeeksi lähelle laskeutumisalueen reunusta, annetaan se anteeksi pelaajalle. Pe-

lin vaikeustason on tarkoitus syntyä syöksyalueiden hyödyntämisestä ja niiden sisällä paikoittautumisesta, ei syöksymisen jälkeen laskeutumisesta. Jos pelaaja on saanut oikean suunnan syöksyessään, pelin tulisi auttaa pelaajaa pääsemään perille seuraavaan osaan, oli se sitten seuraava syöksyalue, seinähyppy tai turvallinen laskeutumisalue. Mikäli pelaaja suoriutuu näistä tempuista, olisi suuri pettymys pelaajalle epäonnistua viimeisessä kohdassa eli laskeutumisessa, sillä vaikeuden tulisi nimenomaan tulla ilmassa tempuilusta. Mekaniikka siis öljyää pelin sekundaarista haastetta huomattavasti helpommaksi. (19) Peli tulkitsee pelaajan sijainnin suhteessa ympäristöön, eli siis kontekstiin, ja muokkaa syötteen vastetta hyvinkin radikaalisti, tuottaen pelaajalle suotuisan tilanteen. Tässä tapauksessa on kyseessä kuitenkin selkeästi pelaajalle huomattavissa oleva pelimekaniikka, eikä näkymätön tempu, sillä näin merkittävää pelaajan sijaintia manipuloivaa mekaniikkaa on hankala piilottaa. Kuva 20 havainnollistaa etäisyyttä, josta pelihahmo voi tarttua tason reunaan.



Kuva 20. Etäisyys, jonka pelihahmon alareuna voi olla tartuttavasta reunasta, jotta pelihahmo voi tarttua siihen.

Kaikki nämä eri tavat yrittävät mukauttaa kulmikasta peliympäristöä pelihahmon kaarimaiseen liikkumistapaan. Niiden tarkoitus on vähentää pelaajan turhautumista kaikennäköisissä rajatapauksissa, joissa ilman näitä muokkauksia peli rankaisisi pelaajaa ja rajoittaisi pelihahmon liikettä. Pelin säännöt — lähinnä sääntö, jonka mukaan pelaaja pysähtyy — kun osuu seinään — ovat OCDassa siis hyvin muovailtavia ja muuttuvat hyvin useasti pelaajan tilasta riippuen. Keinojen on tarkoitus olla näkymättömiä pelaajalle, toimien vain tapauksissa, joissa ero on liian pieni pelaajan

huomattavaksi. Reunaan tarttuminen on kuitenkin OCDAlle uniikki mekaniikka, joka ei ole yleisesti käytössä tasohyppelypeleissä ja on selvästi pelaajalle näkyvässä ja tarkoitettu käytettäväksi. OCDAn pelihahmon liike on siis hyvin öljytty. Pelaajan syötteen tarkoitus yritetään toteuttaa, vaikka pelihahmo ei oikeasti olisikaan oikeassa sijainnissa oikeaan aikaan, jotta haluttu vaste tapahtuisi.

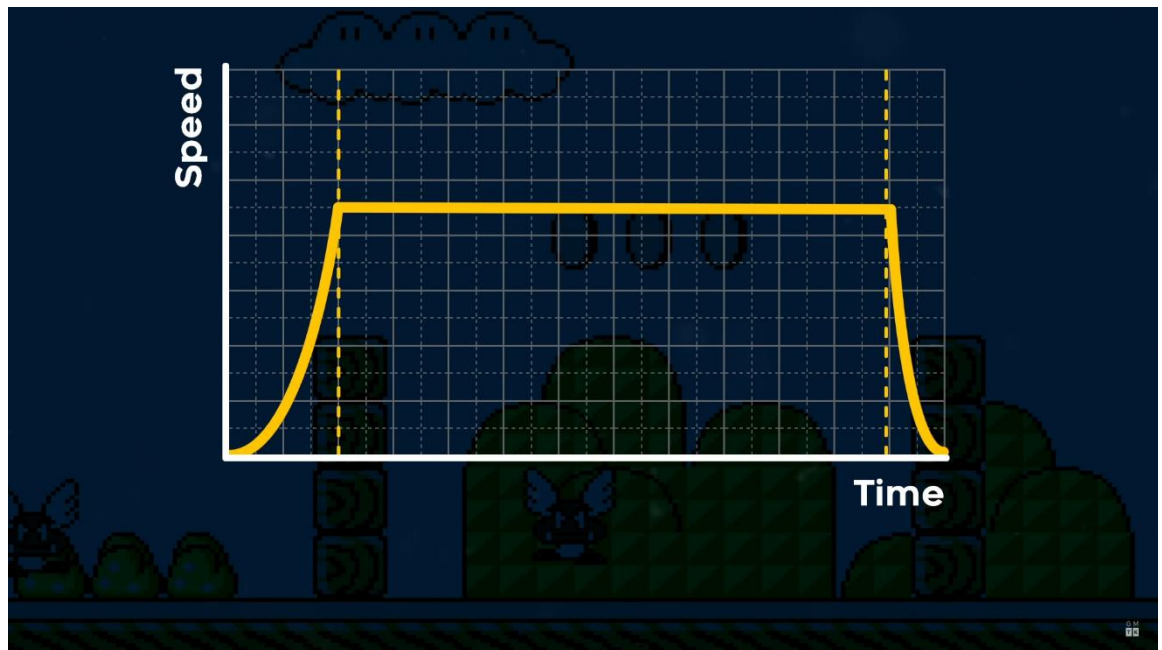
#### 6.4 Pelihahmon attribuutit ja tasosuunnittelu

Tasohyppelyä suunnitellessa tasosuunnittelun pitää myös myötäillä pelihahmon kykyjä. Tasolla voidaan muuttaa kontekstia, mutta tason mitat täytyy suhteuttaa pelihahmon kykyihin. Etenkin tasohyppelypelissä esimerkiksi pelihahmon hyppykorkeus rajoittaa tasosuunnittelun mitataavaa. Yksittäiset hyppyt eivät voi olla pelihahmon hyppykorkeutta suurempia ilman jotain muuta elementtiä, joka auttaa pelihahmoa hypyssä. Muuten pelihahmo ei kykenisi etenemään tasossa. Samoin pelihahmon koko ja hypyn korkeus rajoittavat tason minimiskaalan. Hahmon pitää mahtua liikkumaan tasossa mukavasti ilman, että se törmäilee koko ajan seiniin. Hypyn korkeuden lisäksi pelihahmon liikkumisnopeudella on intiimi yhteys pelihahmon tuntumaan. Pelin vaste pelaajan syötteeseen määrittää, miten pelihahmo kiihtyy ja hidastuu ja mikä on pelihahmon huippunopeus. (1 s. 89) Esimerkiksi hyvin nopeasti huippunopeutensa saavuttava pelihahmo tuntuu paljon helpommalta ohjata kuin hahmo, jolla kestäisi useita sekunteja kiihdyttää huippunopeutensa (21). Hitaammin kiihdyttävällä pelihahmolla voitaisiin esimerkiksi tutkia laajempia tasoja, joissa keskitytään pelihahmon hallitsemiseen laajoissa kaarissa.

Koska OCDA vaatii pelaajalta tarkkuutta pelihahmon sijoituksessa, pelihahmon huippunopeus on pelimaailman kokoon verrattuna pienehkö. Pelihahmo saavuttaa tämän nopeuden nopeasti ja voi vaihtaa kulkusuuntaansa nopeasti ilmassakin, tosin hieman hitaammin kuin maassa kulkien. Vaikka kiihdytys on nopea, sen tuloksena pelaaja voi napauttaa liikkumispainikkeita nopeasti liikkukseen erittäin pieniä etäisyyksiä, hienosäätäen pelihahmon sijaintia, sillä pelihahmo ei ehdi saavuttaa huippunopeuttaan napautuksen aikana (1 s. 128). Ilman tätä pientä kiihtyvyyttä pieninkin painikkeen napautus linkoaisi pelihahmon huippunopeudella painettuun suuntaan. Pienemmällä kiihtyvyydellä voitaisiin taas tehdä pieniä säätöjä helpommin, mutta pelihahmolla kestäisi tämän vuoksi myös enemmän aikaa vaihtaa kulkusuuntaansa.

Pelihahmon hyppy noudattaa samanlaisia sääntöjä. Kun pelihahmo hyppää, pelaaja voi säätää hypyn korkeutta syötteellään. Napautus aiheuttaa pienen hypyn, kun taas hyppypainikkeen poh-

jassa pitäminen johtaa suurempaan loikkaan. Hypyn vertikaali nopeus ja painovoima muodostavat kaaren, jonka suuruus riippuu siitä, kuinka pitkään painiketta pidetään pohjassa. Kun pelihahmo on saavuttamassa hyppynsä huippua, hyppypainikkeen pohjassa pitäminen myös hidastaa pelihahmon putoamista hieman. Yhdistelemällä eri pituisia hyppypainikkeen painalluksia pelaajalla on hyvin paljon valtaa pelihahmon hypyn kaaren muodon muokkaamiseen tilanteen mukaan. Pelihahmon hyppyä suunniteltaessa voidaan ottaa huomioon myös pelihahmon ilmassa viettämä aika, sillä jos pelihahmo on ilmassa yli sekunnin ajan yhden hyppysyötteen tuloksena, alkaa hyppy tuntumaan leijuvalta (29). Kuva 21 visualisoi pelihahmon nopeutta.



Kuva 21. Pelihahmon kiihdytys, maksiminopeus ja hidastuvuus visualisoituna. (21)

OCDAssa pelihahmon hyppy on mitoitettu pelimaailman lohkojen koon mukaisesti. Pelaaja pystyy vaihtelevaan pelihahmon hypyn korkeutta, mutta pelihahmon maksimihyppykorkeus on noin kaksi ja puoli lohkoa. Pelihahmo pystyy helposti hyppäämään ilman mitään apua kahden lohkon korkuiselle tasolle, ja reunukseen tarttumalla kolmen lohkon päälle. Seinähypyn korkeus on sama kuin normaalin hypyn. Syöksyn korkeus suoraan ylöspäin tähdättynä on noin neljä lohkoa ja diagonaalisesti tähdättynä se on kahden lohkon korkuinen. Pelihahmon horisontaalista liikettä hypyjen aikana voidaan mitoitaa samalla tavalla. Käyttämällä lohkojen kokoa mittana voidaan tasoa suunniteltaessa tietää, missä mittakaavassa hyppyjä suunnitellaan (28). Mikäli halutaan pelaajan pääsevän tasolle, joka on kuudentoista lohkon korkeudella, tarvitsee pelihahmo joko syöksyalueita ja/tai seiniä, joita pitkin hyppiä ylös tasoa kohti. Kuvassa 22 OCDAn pelihahmon hypyn korkeus mitattuna lohkoilla.

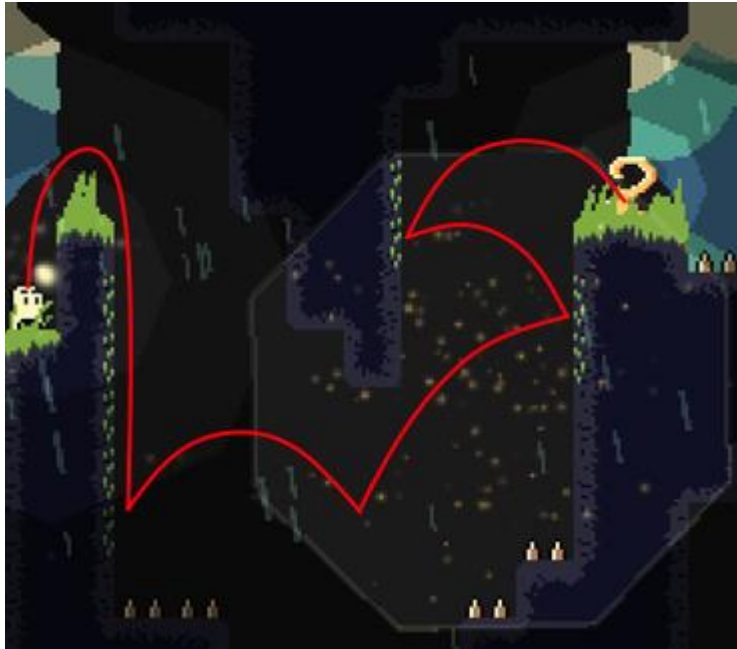




Kuva 22. Pelihahmon hyppy mitattuna lohkojen määrällä.

Jotta pelihahmon luontainen liikerata toteutuisi, yksittäiset hypyt tulisi suunnitella myötäilemään näitä mittoja. Yksittäiset esteet voivat olla näitä mittoja pienempiä, mikäli niihin sisältyy jotain muita haasteita. Kuitenkin täytyy pitää mielessä pelihahmon hypyn ja syöksyn minimipituus. Ei voida suunnitella estettä, jossa pelihahmon täytyy syöksyä ja esteen koko on pienempi kuin syöksyn minimipituus. Mikäli este on suurempi kuin yksittäinen pelihahmon hypyn tai syöksyn pituus, täytyy se suunnitella luontevaksi pelihahmolle. Mikäli este on vain hieman liian suuri pelaajan hypylle, pelaaja saattaa luulla, että pelihahmo pystyy kuitenkin etenemään näin ja yrittää pelata tasoa eteenpäin väärällä tavalla. Yksittäisten esteiden tulisi olla pelaajalle selkeitä. Esteen täytyy olla selvästi joko hypättävissä tai selvästi mahdoton. Suuren esteen valloittaminen vaatii siis monta pienempää estettä pelaajalle. Näiden pienempien esteiden tulisi sitten olla pelihahmon hyppyetäisyyden kokoisia, joita yhdistelemällä pelihahmo voi valloittaa esteen kokonaisuutena. Kun este jaotellaan pienempiin, pelihahmon hypyn mittaisiin esteisiin, on pelaajan helpompi hahmottaa tarkoitettu reitti, jota pitkin edetä tasossa. Tällä tavalla pelaaja voi muodostaa suunnitelman, jonka hän sitten yrittää toteuttaa. Kun uusia esteitä tulee näkyviin, pelaaja jatkaa ja muokkaa suunnitelmaansa lennossa, sen sijaan, että yrittäisi arvata, mitä hänen tulisi tehdä. (30) Koska pelihahmo voi kuitenkin pudota alaspäin niin paljon kuin taso sallii, yksittäiset esteet, joissa pelaajan täytyy pudota alaspäin voivat olla melkein pä millä etäisyydellä tahansa, kunhan ne ovat pelaajan nähtävissä. Näiden yksittäisten esteiden määrä ja laatu määrittävät tason kontekstin kokonaisuutena. Taso, jossa OCDAn pelihahmon ei tarvitse käyttää syöksyä ollenkaan on paljon helpompi ja todennäköisesti vähemmän mielenkiintoinen pelaajalle. Kun taso yhdistää esimerkiksi

pelihahmon hyppyjä, seinähyppyjä ja syöksyä yhtä aikaa, voidaan pelaajalle suunnitella syvempiä ja mielenkiintoisempia esteitä suoritettavaksi. Kuvassa 23 moniesteinen haaste OCDAssa.



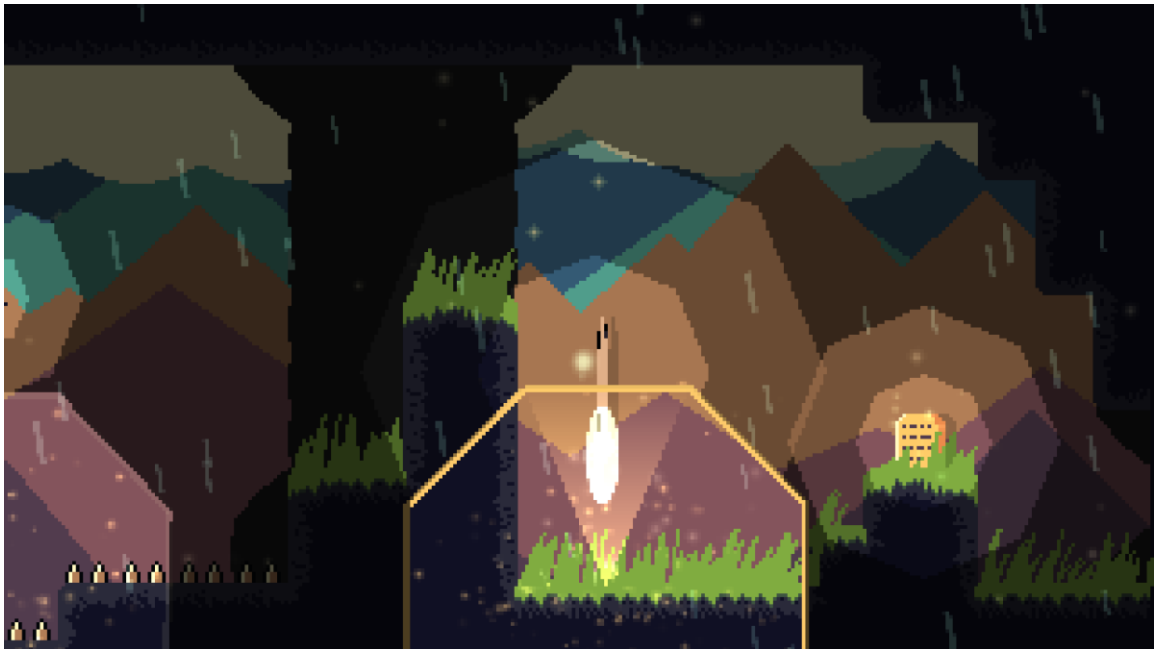
Kuva 23. Moniaskeleinen este OCDAssa, jossa jokainen este on pelihahmon luontaisen hyppy- tai syöksypituuden päässä.

Kun yksittäiset esteet ovat pelihahmon luontaisen liikeradan sisärajalla, pelaaja voi sulavasti liikkua esteeltä esteelle pysähtelemättä matkalla. Koska yksittäiset esteet ovat tarpeeksi pienen etäisyyden päässä, pelaaja voi aina nähdä seuraavan esteen edetessään. Tason lopullisen tavoitteen ei tarvitse välttämättä olla aina näkyvässä, mutta tason ylimalkaisen kulkusuunnan tulisi olla selvä pelaajalle ensimmäisten esteiden perusteella. (8; 31) Tason ulkoreunojen eli katon, lattian ja seinien tulisi yhdessä muodostaa pelaajalle selvä etenemissuunta. Pelihahmolle saatavilla olevaa liikkumistilaa vähentämällä voidaan varmistaa, että pelaaja tietää oikean suunnan.

## 6.5 Pelihahmon viimeistely

OCDAssa pelihahmon viimeistely on verrattain yksinkertainen. Pelihahmolla on yksinkertaiset animaatiokehukset juoksemiselle, hyppäämiselle ja syöksymiselle. Nämä animaatiot ja graafinen ulkoasu ovat olemassa vain sovittaakseen pelihahmoa sitä ympäröivään pelimaailmaan, jotta pelin viimeistely olisi mahdollisimman koherentti. Hahmon viimeistelyssä on kuitenkin käytetty pieniä tehokeinoja, joiden avulla pelihahmosta saadaan paremman tuntuinen muuttamatta mitään toiminnallisuutta.

Pelihahmon animaatiokehkyksiä manipuloidaan litistämällä ja venyttämällä. Kun pelihahmo hypää, se venyy pidemmäksi y-akselilla. Pitääkseen pelihahmon näennäisen massan samana, pelihahmo myös kapenee x-akselilla hypätessä. Pelihahmo näyttää hypyn aikana kurottavansa ylöspäin, kun se litistetään tällä tavalla. Kun pelihahmoa on venytetty, se palautuu pienessä hetkessä takaisin normaaliin skaalaansa. Kaikkia OCDAn pelihahmon verbejä korostetaan tällaisella efektillä. Pelihahmoa venytetään aina, kun sen nopeus vaihtuu huomattavasti, etenkin syöksyessä. Venytys tapahtuu aina liikkeen suuntaan. Jos pelihahmo liikkuu nopeasti ja pysähtyy yhtäkkiä, voidaan pelihahmoa litistää pintaa vasten, mihin se osuu, kuten OCDAn pelihahmolle käy sen laskeutuessa maahan. Nämä efektit antavat pelihahmolle sarjakuvamaisen tuntuman, jokaisessa hypyssä ja syöksyssä tuntuu kuin pelihahmon venyvä keho yrittäisi pysyä liikkeessä mukana. (32) Kuva 24 näyttää tällaisen venytyksen suurimmillaan.

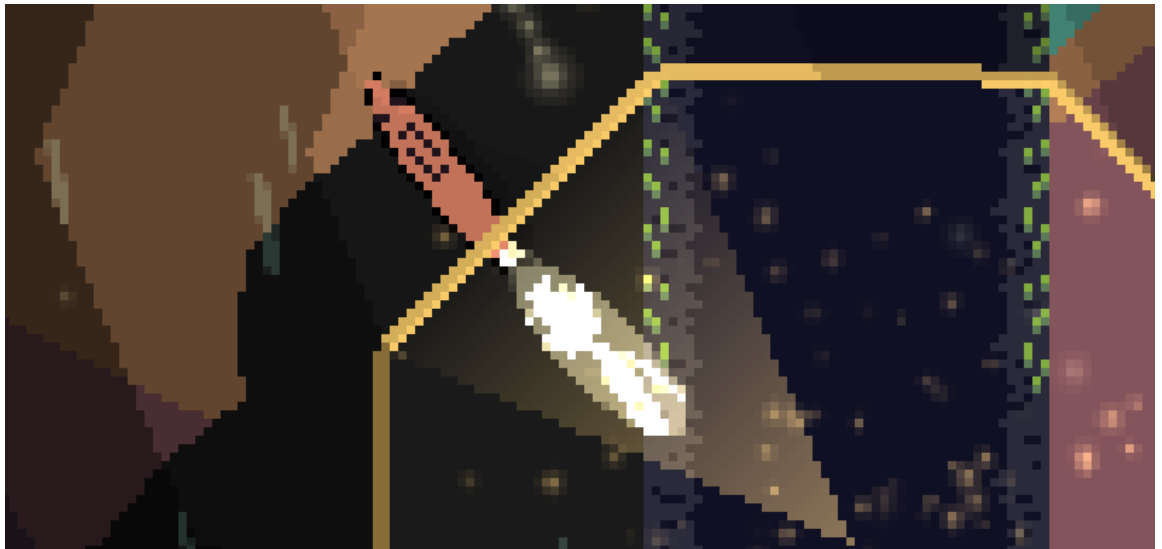


Kuva 24. OCDAn pelihahmon venytys ja litistys ylöspäin syöksyessä.

Pelihahmon kaikkien verbien viimeistelyä täydennetään partikkeliefekteillä. Kun pelihahmo liikkuu, se heittää pieniä pölypartikkeleita ilmaan. Kun pelihahmo vaihtaa liikkumissuuntaa maassa, se myös heittää ylimääräisiä pölypartikkeleita ilmaan kääntymisen vastakkaiseen suuntaan, kuin pelihahmon jalat kuopaisivat maata kääntyessään. Hypätessä ja laskeutuessa päästetään ilmoille pieni pölypilvi. Samoin seinää pitkin liukuessa seinästä irtoaa pölypartikkeleita. Kaikki partikkelit korostavat pelihahmon fyysistä interaktiota ympäristön kanssa. Viimeisenä silauksena pelihahmon liike vaikuttaa sen lähellä oleviin ruohonkorsiin, heiluttaen niitä pelihahmon liikkeessa niiden ohi. Ilman näitä pieniä partikkeleita ja ruohoa pelihahmo tuntuisi liitävän maan päällä il-

man kitkaa, sen liikkeet eivät tuntuisi vaikuttavan ympäristöön (4). Maasta hyppääminen tai maahan laskeutuminen ei tuntuisi vaikuttavan ympäristöön millään tavalla. Viimeistely ei tietenkään vaikuta ympäristöön, mutta pelaajasta se tuntuisi, kuin pelihahmo vaikuttaisi siihen. (10)

Syöksyessä pelihahmon ympärille ja menosuunnan mukaisesti päästetään montaa eri partikkelia samanaikaisesti. Syöksyessään pelaaja ikään kuin ponkaisee takanaan olevasta ilmasta eteenpäin, päästäen abstrakteja partikkeleita taakseen. Lisää partikkeleita päästetään pelihahmon syöksyn suuntaan. Partikkelit maalaavat suoran viivan eteenpäin ja yrittävät auttaa pelaajaa hahmottamaan pelihahmon kulkusuunnan syöksyn jälkeen. Kuvassa 25 näytetään mainitut partikkelit emis-



Kuva 25. OCDAn pelihahmon syöksypartikkelit syöksyn hetkellä.

Jokaisella pelihahmon verbillä on myös oma ääniefektinsä. Koska peli on hyvin retrotyylinen, ääniefektit ovat tätä teemaa mukaillen hyvin yksinkertaisia. Hypyn ääniefekti on klassisista videopeleistä tuttu taajuuskorkeudeltaan nouseva ääni. Kaikenlaisia pelihahmon osumia vahvistetaan myös pätkillä valkoista kohinaa. Kohina on karua kuunneltavaa, mutta pienissä pätkissä kuultuna se vahvistaa pelihahmon osumien tuntua epämiellyttävänä häiritsemättä pelaajaa. Syöksyn ääniefektinä käytetään näiden kahden sekoitusta, jossa valkoisen kohinan taajuuskorkeus nousee kuin hyppääni. Syöksy on ikään kuin hyppy, mutta myös pelihahmolle raju nopeuden muutos, jota kohinan kovuus tuo ilmi.

Ääniefekteille on tärkeintä, että ne sopivat saumattomasti pelin visuaaleihin. Koska OCDAn tyyli ja ylimalkainen metafora on hyvin abstrakti, ääniefektit voivat nauttia samanlaisista vapauksista. Minkään ääniefektin ei tarvitse edustaa mitään oikeaa asiaa tarkasti, vaan voivat vain vahvistaa verbien laatuja. Kun pelihahmon verbillä on sekä graafinen että audiaalinen viimeistely, joiden yhdistelmä on harmoniassa, verbi on pelaajalle miellyttävämpi toteuttaa.

## 7 Paikallinen moninpeli

Chucksaw on 2–4 pelaajan moninpeli, jossa tavoitteena on eliminoida muut pelaajat joko tönäisemällä heidät alas kuiluihin tai osumalla heihin heitettävillä sahoilla. Pelaajat yrittävät ensimmäisenä voittaa kymmenen erää voittaakseen pelin. Kuvassa 26 pelitilanne Chucksawista.



Kuva 26. Neljän pelaajan peli Chucksawissa.

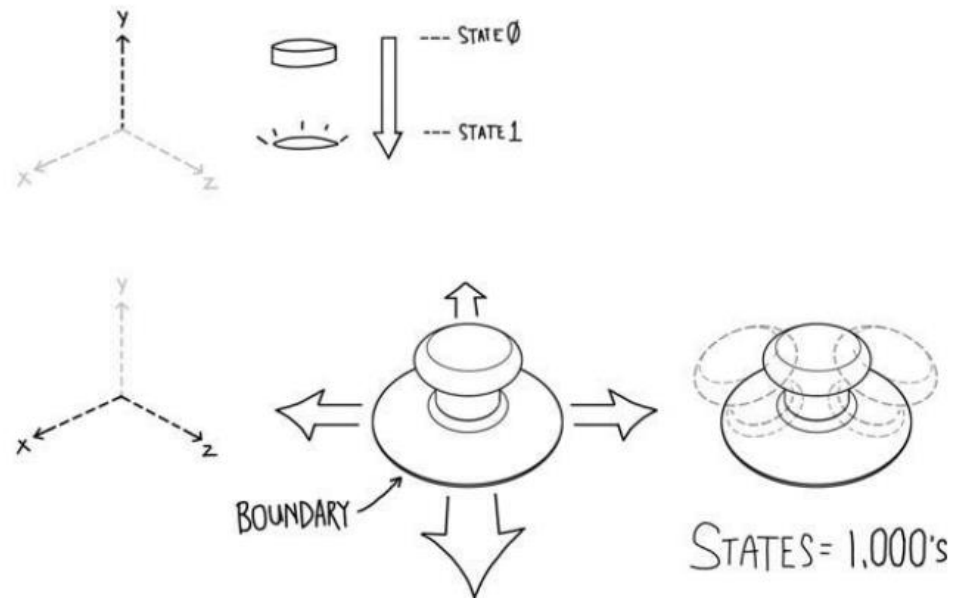
Pelaajat voivat napata sahan, heittää sen tai syöksyä sitä kohti kimmottaen sen. Tämän lisäksi pelaaja voi syöksyä muita pelaajia kohti ja tönäistä heitä. Peliympäristössä on kuiluja, joihin pelihahmot voivat pudota sekä seiniä, mihin sekä pelaajat että sahat törmäävät.

Peli on erittäin nopeampainen, ja usein refleksit eivät riitä pelin tahtiin. Voittaakseen pelaajien täytyy ennustaa muiden pelaajien aikeita ja keksiä vastaisku toista pelaajaa vastaan. Pelin päämekaniikka, sahan heittäminen, on tehty mahdollisimman ekspressiiviseksi, jotta pelaajilla on mahdollisimman monta tapaa pitää vastustajansa arvailemassa.

### 7.1 Yksinkertaisen syötteen kääntäminen monimutkaiseksi vasteeksi

Chucksawin ainoina syöteinä toimivat peliohjaimen ohjaussauva sekä yksi painike. Ohjaussauva lähettää pelille arvoja sen horisontaaliselta ja vertikaalilta akselilta, joista molemmat vaihtelevat arvojen -1 ja 1 välillä desimaalilukuina. Kun ohjaussauvaan ei kosketa, ovat molemmat arvot nolla.

Nappula lähettää pelille yksinkertaisesti signaalin siitä, onko se painettu pohjaan vai ei, yksi tai nolla. Chucksawissa pelihahmo voi kuitenkin suorittaa monia eri toimintoja tästä rajoitetusta syötteestä huolimatta. Kuva 27 visualisoi pelille lähetettyjä arvoja.



Kuva 27. Ohjaussauvan ja painikkeen syötemahdollisuudet. (1 s. 88)

### 7.1.1 Ohjaussauva

Ohjaussauvan syötettä käytetään pelihahmon liikkuttamiseen ja pideltysten saojen tähtäämiseen. Pelihahmon liikkumisen sulavuuteen vaikuttaa paljon itse peliohjain. Ohjaussauvan fyysinen rakenne ikään kuin pakottaa sulavan liikkumisen, sillä akselien arvoja käytetään kertoimena pelihahmon liikkumisnopeuteen. Sauvan sijainti ei voi vaihtua silmänräpäyksessä ääripäästä toiseen, vaan sen täytyy fyysisesti liikkua näiden arvojen välillä. Tuloksena on pelihahmon liike, joka ohjaussauvan tavalla liikkuu sulavasti suunnasta toiseen. Pelaajan ei tarvitse koskaan vapauttaa ohjaussauvaa, vaan voi pitää peukalon sauvan päällä, saaden jatkuvaa palautetta proprioseptisellä tasolla (5). Koska tämä raakasyöte on valmiiksi ikään kuin jo interpoloitu ohjaimen takia, syötteen vaste on hyvin lähellä raakaa syötettä. Kun pelaaja on pelannut hetken, hän osaa aavistaa, mitä syöte tekee liikutellessaan ohjaussauvaa ennen kuin peli ehtii antaa visuaalista palautetta syötteestä.

Ohjain saattaa kuitenkin lähettää pieniä arvoja pelille, vaikka pelaaja ei koskisikaan sauvaan, sillä sauva ei aina välttämättä palaudu täysin keskelle riippuen ohjaimen fyysisestä materiaalista ja

laadusta. Tämä on erittäin yleinen ongelma, ja hyväksi ratkaisuksi on todettu implementoida niin sanottu deadzone eli kuollut alue ohjaussauvan keskellä. (33)

Voidaan valita kuolleen alueen raja, jonka sisällä olevia arvoja ei huomioida ollenkaan. Esimerkkinä voidaan valita rajaksi arvo 0,1. Ohjaussauvan akseleiden itseisarvoa tarkistetaan tätä arvoa vasten ja mikäli akselin itseisarvo on pienempi kuin valittu kuolleen alueen arvo, ei syötettä käytetä ollenkaan. Tarkoituksena on eliminoida erittäin pienet syötteen pelaajalta, sillä näin pieni ohjaussauvan kosketus saattaa hyvinkin olla vahingollinen, eikä pelaaja ole halunnut tehdä sitä tai kokonaan ohjaimen tuottama virhesyöte (33). Mikäli pelaaja on sittenkin halunnut liikuttaa ohjaussauvaa, ja täten halunnut liikuttaa pelihahmoa, tämä voi olla periaatteessa haitallista, sillä pelaajan toive ei toteudu. Todellisuudessa näin pienen syötteen vaikutus pelihahmon liikkumiselle olisi kuitenkin niin pieni, että se ei suurimmassa osassa tapauksista vaikuttaisi pelin tapahtumiin millään tavalla. Kuolleen alueen toteuttaminen siis varmistaa, että vain pelaajan varmasti haluamat syötteen otetaan huomioon pelissä, eikä pelihahmo liiku itsestään pelimaailmassa. Pelin vasteherkkyys on siis pienempi ohjaussauvan kuolleen alueen tuloksena, mutta pelaajan syötteen tarkoituksellisuus on suurempi (22). Mikäli ohjaussauvan syöte on tämän kuolleen alueen ulkopuolella, voidaan sitä käyttää pelihahmon ohjaamiseen.

Kun pelihahmo pitelee sahaa, ohjaussauvan tarkoitus muuttuu. Liikkumisen sijasta sauvan syötettä käytetään sahan heiton tähtäämiseen. Kun pelihahmo päästää sahasta irti, se lentää ohjaussauvan pidettyyn suuntaan. Pelihahmo ei siis pysty liikkumaan samaan aikaan, kun se pitelee sahaa. Koska samaa syötettä käytetään molempiin verbeihin, pelaajan ei tarvitse ikinä liikuttaa sormeaan ohjaussauvalta ja tuloksena sahan nappaaminen tuntuu sulavammalta. Koska pelihahmo on myös täysin paikallaan heittämisen aikana, tähtääminen on pelaajan osalta helpompaa. Tämä antaa myös mahdollisuuden toisille pelaajille liikkua kauemmas sahaa pitelevästä pelaajasta, jotta heillä olisi enemmän aikaa reagoida sahan heittoon.

Kun pelaaja heittää sahaa, hän kasvattaa heittovoimaa heilauttamalla ohjaussauvaa heiton aikana. Heiton voimakkuutta kasvatetaan ohjaussauvan liikenopeuden mukaisesti, kun sahasta päästetään irti. Ongelmaksi syntyy jälleen tarkkuuden vaatiminen pelaajalta. Jotta tätä ohjaussauvan heilautusta voitaisiin helpottaa pelaajalle, voidaan taas hyödyntää puskureita. Kun pelaaja pitelee sahaa, sen sijaan, että peli tallentaisi aina ohjaussauvan nykyisen liikenopeuden, peli tallentaa kaikki sauvan nopeudet pieneltä aikaväliltä, ja valitsee siltä aikaväliltä suurimman nopeuden ja käyttää sitä kaavassa, jossa lasketaan heiton nopeus, kun pelaaja päästää painikkeesta irti. Tuloksena pelaaja voi heilauttaa ohjaussauvaa ja luotettavammin heittää sahan lujempaa. Vaikka

pelaaja heittäisikin sahan hieman ohjaussauvan heilautuksen jälkeen, peli noutaa puskurista heilautuksen nopeimman arvon, ja pelaaja onnistuu heitossaan.

Käänteisesti, mikäli pelaaja päästää sahasta irti liian aikaisin, voidaan jatkaa heilautuksen puskurointia hetken aikaa heiton jälkeen. Mikäli puskurista löytyy suurempi nopeus kuin millä saha heitettiin, voidaan sahan nopeutta päivittää vielä hetken aikaa heiton jälkeen. Myös ohjaussauvan suunta voidaan tarkistaa puskurin ajalta ja tarkistaa, täsmääkö se sahan heittosuuntaan. Mikäli suunnat eroavat toisistaan, peli muuttaa sahan kulkusuuntaa tiettyyn rajaan asti, jotta se täsmäisi paremmin pelaajan syötettä.

Peli siis tarkistaa pelaajan syötteen ennen heittoa sekä heiton jälkeen. Peli mittaa eron pelaajan syötteessä ja heiton hetkellä olevan syötteen välillä ja yrittää sovittaa kaikki kolme instanssia syötteestä yhdeksi koherentiksi tapahtumaksi. Pelaaja pystyy kaiken tämän prosessoinnin takia luottamaan, että heilautukset toimivat ja että saha lentää haluttuun suuntaan jokaisella heitolla.

### 7.1.2 Painike

Vaikka painikkeen raakasyöte on täysin binäärinen, voidaan siitä kuitenkin ekstrapoloida enemmän dataa kuin vain yksi tai nolla. Sen lisäksi että havaitaan, onko painike pohjassa vai ei, voidaan havaita, milloin painikkeen tila muuttuu, eli kun painikkeen syöte vaihtuu nollasta yhteen tai toisinpäin. Kun tähän yhdistetään vielä näiden syötteiden yhdistelmiä, saadaan binäärisestä syötteestä hyvin ekspressiivinen vaste. Painikkeella suoritettavista verbeistä saadaan tehtyä monimuotoisia aktivoimalla ne tällä tavalla. (34)

Pelihahmon verbit eli toiminnot, joita pelaaja pystyy suorittamaan painikkeella, ovat sahan nappaaminen, sahan piteleminen, sahan heittäminen ja syöksyminen. Kaikki paitsi syöksyminen ovat tilannekohtaisia verbejä, eli ne riippuvat jostain pelihahmon ulkopuolisesta muuttujasta toimiakseen. Tässä tapauksessa ne riippuvat sahan läheisyydestä pelihahmoon. Kaikki verbit jakavat myös saman resurssin pelihahmossa. Kun pelaaja on suorittanut yhden toiminnon, täytyy pelaajan odottaa hetki, että resurssi tulee takaisin ennen kuin hän voi suorittaa näitä verbejä uudelleen. Verbejä on enemmän kuin nappulan raakasyöte sallii ja pelaaja voi siirtyä yhdestä verbistä toiseen sulavasti, joten syötettä täytyy moduloida.

Sahan napatakseen pelaajan täytyy painaa painiketta, kun saha on tarpeeksi lähellä pelihahmoa. Konseptina tämä on yksinkertainen, mutta sahan liikkeessa hyvin nopeasti ja vaihtaen kulkusuuntaa törmäyksen tai toisen pelaajan heiton takia se voi olla hyvin vaikeaa käytännössä. Pelin tilanne



voi vaihtua nopeammin, kuin mitä ihmisen havainnointisykli sallii pelaajan havaita. Mikäli tämä verbi tapahtuisi vain silloin, kun painikkeen tila muuttuu ei-painetusta painettuun, sahan nappaaminen olisi melkein mahdotonta.

Kun pelaaja painaa painikkeen ei-painetusta tilasta painettuun tilaan, pelaajalla on tietyn pituinen syötepuskuri, jonka aikana pelaaja voi silti napata sahan. Sen sijaan, että pelaajan täytyisi painaa painiketta tismalleen oikealla hetkellä, joka saattaa kestää vain muutamia kehyksiä, riittää, että pelaaja pystyy suurin piirtein arvaamaan, milloin saha voisi tulla tarpeeksi lähelle pelihahmoa. Tämä puskuri kestää noin puoli sekuntia ja on samaan aikaan tarpeeksi pitkä, että pelaajan ei tarvitse olla täysin tarkka syötteensä kanssa, mutta samaan aikaan tarpeeksi lyhyt, että pelaaja ei voi vain pitää painiketta pohjassa sokeasti vaan joutuu ennustamaan milloin toinen pelaaja saattaa heittää sahan pelihahmoaan kohti. Mikäli pelaaja pitää painiketta pohjassa ja tämä aika tulee täyteen, hänen pelihahmonsa menettää resurssinsa ja joutuu odottamaan hetken aikaa ennen kuin voi suorittaa näitä verbejä uudestaan.

Jos pelaaja onnistuu nappaamaan sahan, painikkeen tarkoitus muuttuu. Tässä pelihahmon tilassa painikkeen pohjassa pito sahan nappaamisen sijasta jatkaa nykyisestä sahasta pitelemistä, eikä nappaa enää uusia sahoja. Samalla painikkeen pohjassa pitäminen nostaa sahan heittovoimaa, mitä kauemmin painiketta pidetään pohjassa. Samoin kuin sahan nappaamisessa, aika jonka pelaaja voi viettää tässä tilassa, on rajallinen. Mikäli pelaaja ei päästä painikkeesta irti ennen tätä aikaa, hänen pelihahmonsa kuolee. Jos pelaaja kuitenkin päästää painikkeesta irti ennen tätä aikarajaa, hän heittää pitelemänsä sahan ohjaussauvan suuntaan.

Koska painikkeen pohjassa pito aktivoi sahan nappaamisen, pelaajan syöksyminen ei voi tapahtua samasta syötteestä, mikä olisi normaalisti loogisin tapa tehdä se. Sen sijaan syöksyminen tapahtuu, kun pelaaja päästää irti painikkeesta ja ei pitele sahaa. Sama puskuri on kuitenkin edelleen voimassa, eli painikkeesta pitää päästää irti sen aikana, että pelihahmo voi suorittaa syöksyn. Teoriassa tämä aiheuttaa viivettä syötteeseen, sillä pelaajan täytyy sekä painaa painiketta että päästää irti suorittaakseen tämän verbin, mutta käytännössä viive on niin pieni, että pelatessa pelaaja ei sitä huomaa. Samaan aikaan tämä mahdollistaa erittäin sulavan siirtymisen verbistä toiseen käyttämällä rajoitettua syötettä luovasti. Jos pelaaja haluaa yrittää napata sahan, mutta muuttaa mielensä, hän voi puskurin aikana päästää painikkeesta irti ja suorittaa syöksyn sahan nappaamisen sijaan.

Syötteen yksinkertaisuudesta syntyy kuitenkin ongelma. Mikäli pelaaja on tarpeeksi lähellä sahaa napatakseen sen, mutta haluaa syöksyä sahaan sen nappaamisen sijasta, painaessaan painiketta, hän alkaa pitelemään sahaa. Kun pelaaja sitten päästää painikkeesta irti, heittää hän sahan sen

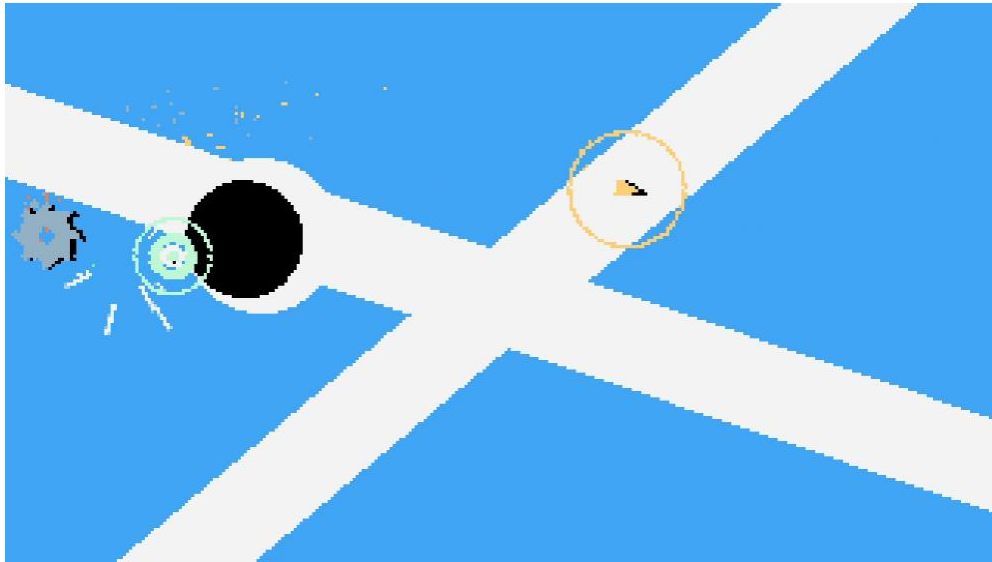
sijaan, että syöksyisi siihen. Jotta pelaajalla olisi valta päättää pelihahmonsa tekemisestä tässä tilanteessa, voidaan implementoida toinen syötepuskuri.

Koska pelaajan suorittama syöte syöksyessä on yleensä hyvin nopea painikkeen painallus, voidaan tätä hyödyntää syötepuskuria suunnitellessa. Kun pelaaja nappaa sahasta kiinni, aloitetaan uusi pusku. Jos pelaaja siis nappaa sahasta kiinni ja päästää painikkeesta heti sen jälkeen irti, peli rekisteröi syötteen syöksynä eikä sahan heittona. Sama syöte ilman tätä puskuria suorittaisi eri verbin. Yksinkertaista puskuria käyttäen peli siis muuttaa tulkintaansa pelaajan syötteestä, yksinkertainen syöte tulkitaan monimutkaisena vasteena.

## 7.2 Audiovisuaaliset efektit kommunikaationa

Kuten Chucksawin pelitahti, pelin sisäiset audiovisuaaliset efektit ovat hyvin nopeita, osa niistä kestää vain muutaman kehyksen verran. Pelaaja ei välttämättä tiedosta kaikkia efektejä, mutta ne silti osallistuvat pelin ylimalkaiseen tuntumaan. Tämän lisäksi efektit on melkein aina väritetty helposti tunnistettaviin pelihahmojen väreihin. Kuten juississa määriteltiin aikaisemmin, efektit ovat aina osana pelitapahtumaa, korostamassa sitä. Tarkemmin tarkasteltuna Chucksawin visuaalinen ulostulo on hyvin pitkälti pelkästään näitä efektejä, juissia.

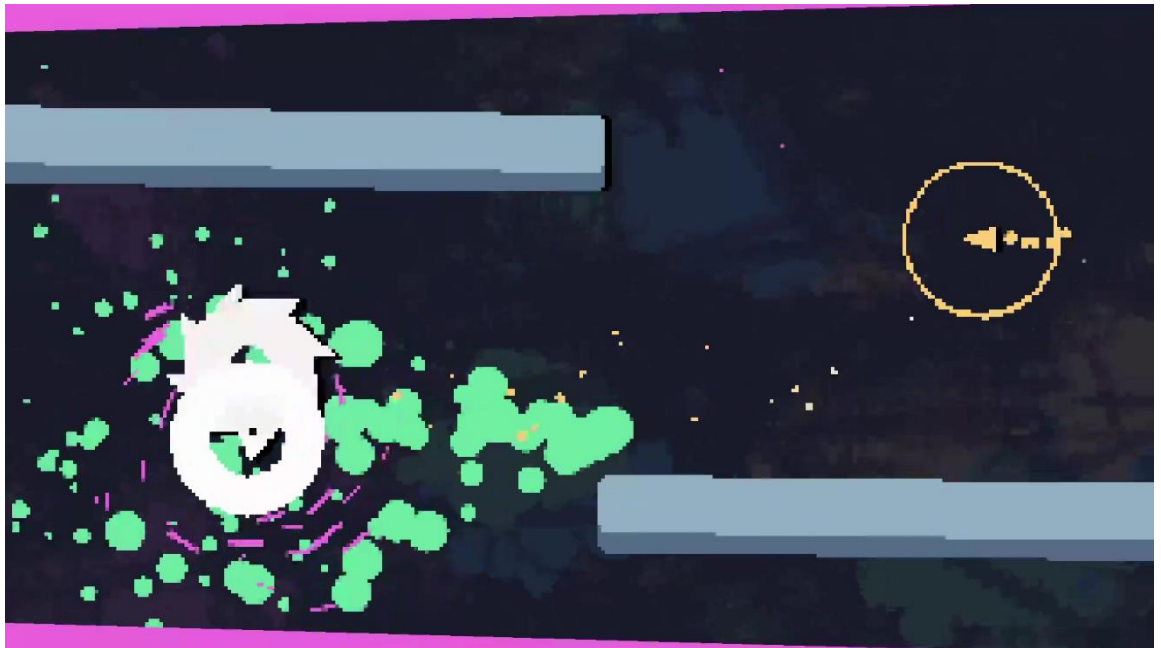
Efektit skaalautuvat yleensä pelitapahtuman tärkeyden mukaan. Pelin tärkeimmiksi tapahtumiksi voitaisiin luokitella pelihahmon kuoleminen, sahan nappaaminen, sahan heittäminen ja sahaan syöksyminen, ja siksi näihin tapahtumiin onkin liitetty huomattavimmat efektit. Näistä tapahtumista merkittävin on kuitenkin pelaajan kuolema. Kun tämä tapahtuu, koko peliruutu täyttyy efekteistä. Kuvassa 28 kuvaa tätä efektiä.



Kuva 28. Pelihahmon kuolinefekti peittää koko ruudun vain muutaman kehyksen ajan. Tärkeimmät peliobjektit kuitenkin piirretään aina päällimmäiseksi ruudulla.

Peliruutu täytetään kuolevan pelaajan värillä, jota vasten korostuu mustavalkoinen rasti ja ympyrä kuolleen pelaajan kohdalla. Koska peli on erittäin hektinen, pelaaja voi olla täysin keskittynyt omaan pelaamiseensa/selviytymiseensä, eikä välttämättä aina tiedä muiden pelihahmojen tilaa. Kuolemaefekti on tämän takia suunniteltu nappaamaan kaikkien pelaajien huomio, sillä heidän tulee olla tietoisia tästä tapahtumasta ilman, että he joutuvat tietoisesti etsimään tätä tietoa pelatessa (4). Efekti pyrkii kertomaan myös muuta tietoa tapahtumasta. Väri tarkoittaa kenen pelihahmo on kuollut ja ympyrän ja rastin sijainti kertoo missä pelihahmo on kuollut. Tämän lisäksi efekti on yksi harvoista, jolla on myös suoranainen vaikutus pelin toimintaan, sillä se pysäyttää pelin kokonaan hetkeksi. Pelin pysähtyminen antaa painoa tapahtumalle ja sen kesto on tarpeeksi lyhyt, että pelaajat eivät ärsyynny siitä. Lisäksi se antaa arvokasta aikaa pelaajille prosessoida, mitä on juuri tapahtunut ja miten se voi vaikuttaa heidän seuraaviin toimintoihinsa pelissä. Nopea, iskevä ja kovaääninen ääniefekti kruunaa pelaajan kuolinhetken, jota seuraa korkeasävyinen särinä, joka vaimentuu hiljattain pelin ollessa pysähdyksissä.

Koko ruudun efektin jälkeen kuollut pelaaja jättää jälkeensä suuren partikkelivanan, joka katoaa hiljalleen. Efekti korostaa sahan osuman voimaa ja partikkelit lentävät siihen suuntaan, josta saha tappoi pelaajan, antaen hieman ylimääräistä tietoa siitä, mikä sahan liikerata on sillä hetkellä. Tämän lisäksi pelaajan kuolema maalaa pelimaailman lattiaan pelaajan värisen jäljen, joka on näkyvissä ottelusta toiseen. Efekti kertoo sekä lattian että pelihahmojen laadusta. Tämä viimeistely auttaa myymään pelimaailmaa fyysisenä, sillä pelaajat voivat jättää jälkensä sinne (2). Kuvassa 29 näytetään pelihahmon kuoleman partikkeliefekti.



Kuva 29. Kuolleen pelaajan partikkelivana.

Pelin toiseksi näyttävin efekti on sahaan syöksyminen. Periaatteessa olisi järkevää ajatella, että sahan heittäminen olisi toiseksi tärkein, koska se on yleisin toiminto, mutta koska sahaan syöksyminen vaikuttaa merkittävästi sekä sahaan että syöksyvään pelaajaan, on siihen liitetty enemmän efektejä. Kun pelaaja syöksyy sahaan, se kimpoaa pelaajasta aina maksiminopeudella. Tämän lisäksi syöksyvä pelaaja taintuu osumasta hetken ajaksi.

Syöksyn osuessa sahaan pelaajan ympärillä oleva ympyrä kasvaa räjähdysmäisesti ja sitten häviää pelaajan ympäriltä. Efekti on hyvin suuri, liikkuu nopeasti ja tarkoituksena on, että muutkin pelaajat huomaavat tämän pelaajan syöksyneen. Osuman kohtaan piirretään parin kehyksen ajaksi myös mustasta valkoiseen välähtävä pallo imitoimassa pientä räjähdystä. Loppusilauksena on lisätty kipinöitä muistuttavia partikkeleita, sillä sahasta yritetään antaa metallinen vaikutelma, oli se minkä värinen tahansa. Kipinät ovat tässä toiminnassa ainoa ei-abstrakti efekti, joka muistuttaa toiminnan fyysikaalisuudesta. Syöksyn implikoitu osuma on helpompi ymmärtää, kun pelissä tapahtuu jotain, mitä pelaaja voi verrata oikeaan maailmaan. Kuvassa 30 pelihahmo hetkenä, kun sen syöksy osuu sahaan.



Kuva 30. Pelihahmo syöksymässä sahaan.

Ääniefektinä toimii muihin sahaan liittyviin ääniin kontrastoiva korkeasävyinen väpähtävä kalahdus. Muiden pelihahmon verbien ääniefektit ovat matalia sävyiltään, joten tälle tärkeälle verbille tulee antaa helposti eroteltavissa oleva ääniefekti, jotta pelaajat voivat reagoida siihen helpommin.

Kun pelihahmo nappaa sahan, sen ja pelihahmon ympärillä olevan ringin väri muuttuu valkoiseksi. Nappauksen hetkellä soi myös napakka ääniefekti sahan onnistuneen nappauksen merkiksi. Peli myös pysähtyy aivan pieneksi hetkeksi sahan nappaamisen hetkellä.

Pelaajan jatkaessa sahan pitelemistä alkaa saha, ringi, pelihahmo sekä pelaajan fyysinen ohjain täriseämään nousevalla voimalla. Tärinän tarkoitus on implikoida pelihahmon käyttävän kaikkia voimiaan sahan pitelemiseen ja heittämiseen. Tärinän kasvaessa myös ringin sisempi reuna alkaa liikkumaan ringin keskustaa kohti, jota pelaaja voi käyttää mittarina sille, kuinka kauan hän voi pidellä sahaa. Tämän lisäksi pelihahmon ympärillä pyörii vielä partikkelimyrsky, joka antaa väriä pelihahmon ympäristölle. Pelihahmon pidellessä sahaa soitetaan myös ääniefekti, joka nousee sävelkorkeudeltaan pelaajan pidellessä sahaa. Yhdessä ringin sisenemisen ja tärinän kanssa nämä efektit luovat yhdessä kasvavan paineen tunteen pelaajalle hänen ladatessa sahan heittoa. Kaikki verbin viimeistelyefektit vahvistavat yhtä tunnetta. (12)

Sahan heitolla itsessään ei ole paljon efektejä. Peli jälleen kerran pysähtyy erittäin pieneksi hetkeksi heiton hetkellä ja samankaltainen ääniefekti kuin sahaa napatessa soitetaan. Mikäli pelaaja onnistuu heittämään sahan tarpeeksi lujaa, soitetaan ylimääräinen ääniefekti ja liitetään mukaan

ylimääräinen partikkeliefekti. Tällaisilla ylimääräisillä efekteillä voidaan kertoa pelaajalle suoraan hänen onnistumisestaan jostain pelin sisäisestä tapahtumasta. Pelaaja saa peliltä varmistuksen onnistumisestaan ilman, että pelin täytyy kertoa sitä suoraan pelaajalle ohjetekstinä. Pelkästään tyydyttävämpi viimeistelyefekti pelihahmon onnistuessa täydellisesti voi motivoida pelaajaa suorittamaan kyseistä verbiä paremmin (13).

Chucksawin efektit kohdistuvat lähinnä pelihahmojen verbien efekteihin eli juissiin, mutta pelihahmon kyky, tai kyvyttömyys, napata saha on myös tärkeä asia kommunikoida. Pelihahmon ympärillä oleva ympyrä kertoo hahmon kyvystä napata saha tai syöksyä. Kun pelaaja tekee jommankumman ja pelihahmo menettää kykynsä vaikuttaa sahaan, rinki kutistuu näkymättömän pieneksi merkiksi siitä, että pelihahmo ei kykene nappaamaan sahaa. Kun pelihahmo kykenee jälleen vaikuttamaan sahaan, rinki kasvaa normaalin kokoiseksi ja pieni partikkeliefekti antaa lisää liikettä tälle tapahtumalle, että se olisi mahdollisimman helppo huomata.

Tähän tapahtumaan voisi lisätä ääniefektin, mutta koska se ei ole pelaajan aktiivinen verbi, vaan tapahtuma, joka tapahtuu itsenäisesti aina tietyn ajan jälkeen, se ei olisi välttämättä viisasta. Koska pelissä on useita pelaajia pelaamassa samaan aikaan, monen ääniefektin soiminen samaan aikaan olisi häiritsevää pelaajille, koska äänillä ei ole selvää lähdettä. Vaikka tapahtuma onkin melko tärkeä pelin sisällä, pelaajat voisivat epähuomiossa vaikkapa luulla ääniefektin perusteella, että heidän oma resurssinsa on tullut takaisin, vaikka se onkin todellisuudessa ollut toisen pelihahmon resurssi, joka on päästänyt ääniefektin.

Saha on pelin sisällä aina liikkeessä. Jatkuvan pyörimisen lisäksi se myös koko ajan päästää kipinöitä muistuttavia partikkeleita sekä savupartikkeleita. Savupartikkelit pysyvät pelimaailmassa pidemmän aikaa ja antavat ikään kuin hännän sahalle. Kun saha saavuttaa ruudun reunan ja siirtyy vastakkaiselle puolelle, pelaajat voivat savuhännän avulla päätellä sahan liikeradan helpommin, vaikka se ei olisikaan vielä näkyvissä (35). Sahaan liitetyt efektit ovat kaikki värikkäitä ja nopeasti liikkuvia, sillä se on epäilemättä tärkein peliobjekti, mihin pelaajien pitää aina kiinnittää huomiota ja tietää, missä se liikkuu. Kuvassa 31 näytetään sahan savuhäntä.



Kuva 31. Sahan sininen partikkelivana, jonka avulla pelaajat voivat päätellä sahan sijainnin ruudun ulkopuolella.

Jotta saha tuntuisi dynaamisemmalta pelimaailmassa, sen pyörimisnopeus on sidottu sen nopeuteen, eli mitä nopeammin se liikkuu, sitä nopeammin se myös pyörii. Mikäli saha liikkuu tarpeeksi lujaa, se myös päästää ylimääräistä tuulipartikkeleita niin kuin se halkaisisi edessään olevaa ilmaa. Kun saha törmää seinään, päästää se myös ylimääräisen kipinäpartikkeliräjähdyksen, jotta se tuntuisi metalliselta.

### 7.3 Kameran tehtävä ja kameraefektien tehokeinot

Yksi Chucksawin suurimpia tehokeinoja välittää tapahtumien kineettisyyttä on erilaiset tavat manipuloida kameran sijaintia, minkä kautta pelimaailmaa tarkastellaan. Kamera on ei-diegeettinen – eli ei pelimaailmaan kuuluva – peliobjekti, joten sen efektit voivat kuitenkin helposti jäädä tiedostamatta pelaajalta (36). Pelaaja siis tuntee kameran vaikutuksen, mutta ei samassa hetkessä välttämättä osaa osoittaa sitä. Kamera on pelin taikapiirin ulkopuolella ja on siis olemassa jossain kohtaa pelaajan ja tietokoneen välissä vuorovaikutteisuuden kaaviossa. Pelaaja havaitsee kaiken pelin sisällä tapahtuvan kyseisen kameran kautta ilman, että itse kamera osallistuu mihinkään pelin tapahtumaan; pelihahmot eivät tiedosta kameran olemassaoloa. Voidaan kuitenkin luoda illuusio siitä, että kamera olisi jollain tavalla osa pelimaailmaa laittamalla se reagoimaan pelin tapahtumiin. Kun kaksi peliobjektia on vuorovaikutuksessa keskenään, voidaan kameraa manipuloida niin kuin se olisi mukana tässä vuorovaikutuksessa.

2D-peleissä yleisimmät kameran sijaintia manipuloivat efektit ovat kameran värinä ja kameran näkymän koon muuttaminen. Tämän lisäksi kameraa voidaan pyörittää tai työntää hallitusti tiettyyn suuntaan (2). Kameran sijainnin lisäksi voidaan manipuloida useita muita kameran ominaisuuksia, kuten vaikkapa fokusta ja jälkikäsitteleyefektejä, mutta Chucksaw käyttää vain kameran sijaintia ja rotaatiota manipuloivia tehokeinoja. Kameraefektit ovat kuitenkin suuri osa pelin viimeistelyä.

Tehokeinot on toteutettu yksinkertaisesti muuttamalla kameran koordinaatteja ja rotaatiota pelimaailman sisällä. Kuitenkin sen sijaan, että kamera liikkuisi jokaisella kehyksellä täysin koordinaattien osoittamaan sijaintiin, kameran liike on interpoloitu sen alkuperäisestä sijainnista kohdekoordinaatteihin. Tuloksena on sulavampi kameran liike, joka kuitenkin pystyy terävähköisiin liikkeisiin. Interpolaation ansiosta kamera myös palautuu sulavasti alkuperäisiin koordinaatteihinsa efektien toteutuksen jälkeen ja estää mahdollisesti suuret kameran heilahdukset, jotta pelihahmot ovat mahdollisimman usein näkyvissä. (37; 38) Tietyllä tavalla tällainen interpolaatio imitoi kameramiestä, joka yrittää pitää kameraa pystyssä tasaisesti osoitettuna kohteeseen, kun joku tönäisee häntä. Kameran tehtävänä on loppujen lopuksi yrittää antaa pelaajalle infoa pelin tilasta mahdollisimman selkeästi ja pelaajan tulisi aina tietää, missä hänen pelihahmonsa sijaitsee. (19; 38) Koska Chucksawissa pelaajia on monta, toiminta on nopeaa ja pelaajalle tärkeä info voi olla missä tahansa tasoa, kamera on ankkuroitu tason keskipisteeseen. Vaikka kameraa siirrellään paljon, yritetään kamera keskittää aina takaisin kaikkien näiden tärkeiden elementtien keskipisteeseen, jotta pelaajilla on mahdollisimman paljon tietoa ympäristöstään (37). Koska peliobjektit voivat liikkua ruudun laidalta toiseen reunojen kautta, ideaalisesti tason reunat olisivat aina näkyvissä.

### 7.3.1 Käytetyt kameraefektit

Chucksaw hyödyntää kaikkia edellä mainittuja sijainnin manipulaation tehokeinoja. Kaikista yleiskäyttöisin funktio on kuitenkin kameran värinä. Kameran värinä korostaa törmäyksen efektiä, koska se luo vertauskuvan oikeaan maailmaan, jossa suuren esineen törmäys vaikuttaisi kameran sijaintiin. Kameraa pitävä henkilö tai jalusta luultavimmin liikkuisi tai kaatuisi, mikäli muutaman tonnin painoinen kivi tiputettaisiin sen viereen korkealta. Useimmiten kameran värinää käytetään räjähdysefekteissä, joissa kameran värinä simuloi räjähdyspaineaaltoa, joka ravistaisi kameraa. Efekti pelin sisällä ei täsmää täysin oikeaan maailmaan, mutta yhteys on selvä pelaajalle. Tämän yhteyden tajuminen kasvattaa peliobjektin painon tunnetta pelaajan silmissä, koska sillä on havaittavissa oleva vaikutus kameraan, jonka kautta tarkastelemme peliä.

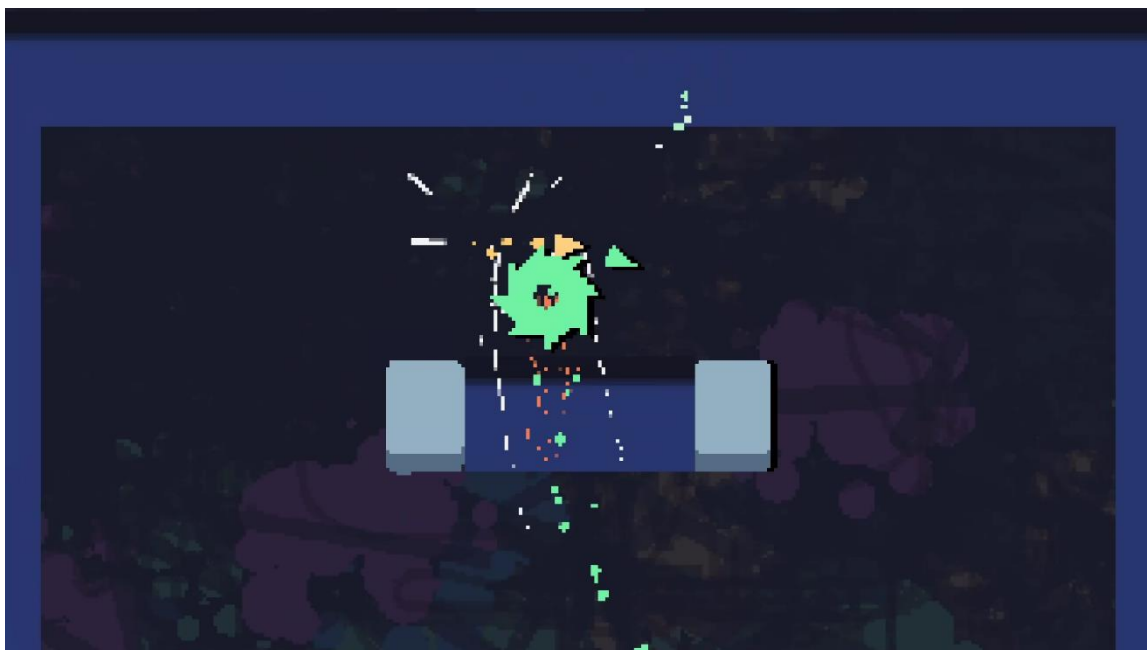


Kameran tärinä voi korostaa mitä tahansa tapahtumaa pelin sisällä, missä objektit törmäävät toisiinsa. Koska kameran tärinällä ei ole varsinaista suuntaa, voidaan sitä käyttää minkä suuntaisissa törmäyksissä tahansa (2). Tärinälle voidaan antaa kaksi muuttujaa: tärinän kesto ja suuruus. Näitä säätämällä tärinälle voidaan antaa erilaisia vivahteita. Nyrkin lyönti voisi aiheuttaa lyhyen, mutta suuren tärinän, kun taas suuri räjähdys kaukaa koettuna aiheuttaisi paljon pidempikestoisen tärinän, joka olisi hellempi. Kun kameran sijainti ja tärinän määrä vielä interpoloidaan, tärinä tuntuu vähemmän ärsyttävältä pelaajalle, koska kameran liike ei ole täysin satunnaista. Tärinän kaaoksen seassa löytyy tietynlainen jatkumo kameran liikkeessä (38).

Chucksawissa melkein jokaista pelaajan toimintoa sahan kanssa korostetaan kameran tärinällä. Kun pelihahmo nappaa sahan ilmasta, kamera tärähtää terävän lyhyesti, simuloiden sahan inertiaa. Saha pidellessä kamera tärisee jatkuvasti suurenevilla voimakkuudella, mitä pidemmin pelihahmo pitelee sahaa. Mikäli pelihahmo pitelee sahaa liian kauan ja räjähtää, kamera tärisee rajusti ja kauan. Samanlainen suuri tärinä tapahtuu aina kun pelihahmo kuolee sahalle. Myös sahan törmäilyä seiniin korostetaan pienellä kameran tärinällä. Tärinän suuruus edellä mainituissa efekteissä myös kasvaa sahan koon ja nopeuden kasvaessa, sillä on järkeenkäypää, että objektin kasvaessa myös sen vaikutus kameraan kasvaisi.

Toiseksi useimmiten Chucksawissa käytetty kameraefekti on kameran työntäminen. Sitä käytetään melkein aina kameran tärisyttämisen kanssa. Kun tärinä luo ylimalkaista törmäyksen tunnetta ja antaa näennäistä painoa peliobjektille, työntö korostaa peliobjektin liikkeen suuntaa ja nopeutta. Samalla tavalla kuin räjähdysten paineaalto puskisi kameraa pois päin räjähdysten keskipisteestä, kameraa työnnetään törmäävän peliobjektin aikaisempaan kulkusuuntaan.

Sahan törmäily seiniin liikuttaa kameraa sahan kulkusuuntaan tärinän lisäksi. Sama tapahtuu pelaajan ottaessa sahaa kiinni tai sitä heittäessä. Kamera työntyy sahan liikeradan mukaisesti, kuin saha työntäisi pelihahmoa taaksepäin sen voimalla. Samalla tavalla sahaa heittäessä kamera työntyy heiton suuntaan kuten pelihahmon vartalon painopiste siirtyisi eteenpäin raskasta esinettä heittäessä. Pelaajan kuollessa sahalle kamera ei enää työnnykään yhtä paljon, sillä on tarkoitus vahvistaa kuvaa sahan leikkaavan läpi pelihahmosta. Kuvassa 32 havainnollistetaan hetkeä, jolloin kameran efektit voivat pinoutua estäen koko tason näkemisen.



Kuva 32. Kameran sijainti on työntynyt huomattavasti tason keskipisteestä.

Myös pelaajan syöksyn osuessa sahan kamera työntyy, mutta tällä kertaa koska pelaajan syöksyn liike on tärkeässä asemassa, kameran työntösuunta on pelihahmon syöksyn liikeradan suuntainen, eikä sahan. Mikäli kaksi peliobjektia törmää ja molemmat ovat liikkeessä, yleensä on selkeämpää valita näistä kahdesta vain toinen, jonka toimintaa korostetaan. Mikäli kaksi yhtä suurta ja nopeaa peliobjektia törmäisi eri suunnista, niiden suuntien summattu vektori voisi olla hankala ymmärtää pelaajalle törmäyksen lyhyessä hetkessä. Useimmiten selkeintä pelaajalle on valita pelihahmon toiminnan suunta ja voima, sillä pelaajalla on tarkempi odotus oman toimintansa tuloksesta, kun taas toisen peliobjektin tila toiminnan jälkeen saattaa olla epäselvempi. Mikäli kaksi peliobjektia törmää täysin vastakkaisista suunnista, luontaisin tulos voisi kuitenkin olla työntämättä kameraa. Tällaisissa tapauksissa kameran työntösuunta ja suuruus on kuitenkin makuasia ja riippuu efektin halutusta tarkoituksesta.

Viimeisiksi efekteiksi jäävät kameran kääntäminen ja näkymän koon muuttaminen. Toisin kuin kameran värinä ja työntö, näillä kahdella efektillä ei ole yhtä helposti olemassa olevaa vertauskuvaa. Tämän takia näiden efektien käyttäminen pelissä on hyvin pitkälti tyylillinen valinta, kun taas kameran värinä on melkein jokaisessa pelissä läsnä oleva efekti, jolla on helposti ymmärrettävissä oleva vertauskuva oikeassa maailmassa.

Kameran kääntäminen on varattu Chucksawissa vain kouralliselle toimintoja, pelaajan kuolemiselle, toisen pelaajan työntämiselle sekä sahan pitelemiselle. Kameran kääntämistä käytetään lähinnä nopeiden liikkeiden korostamisessa, joissa ei muuten olisi paljon inertiaa tai painon tuntua. Kun pelaaja kuolee, leikkaava saha ei työntä kameraa paljon, joten vaihtoehtoinen tehokeino on

kääntää kameraa terävällä liikkeellä, ikään kuin veitsen leikkaavalla liikkeellä. Vertauskuva on vaikeampi yhdistää pelatessa, mutta tärkeintä on, että kamera liikkuu jollain eri tavalla tällaisen tärkeän pelitapahtuman tapahtuessa. Kääntö erottaa tämän tapahtuman muista tapahtumista, joissa kamera työntyy ja tärisee. Kun pelissä tapahtuu hyvin paljon visuaalisia efektejä, kameraefektit auttavat kertomaan pelin tapahtumista kaikkien muiden efektien läpi.

Koska pelihahmojen ei ole tarkoitus tuntua painavilta, kamera lähinnä kääntyy työntymisen sijaan pelihahmon syöksyessä toista pelihahmoa kohti. Tarkoituksena on korostaa osumaa ilman, että se tuntuisi erityisen painavalta. Samoin kuin pelaajan kuolemaefektissä, kamera napsahtaa viinon kulmaan, josta se hitaasti palaa takaisin alkuperäiseen rotaatioonsa.

Pelaajan pidellessä sahaa kamera tärisee sekä kääntyy. Koska kameran kääntyminen on selkeä kuoleman efekti, on kameran kääntymisellä edestakaisin tarkoitus ikään kuin varoittaa siitä, että jos sahaa pitelee liian kauan, se tappaa pelihahmon. Tämän efektin vaikutus on kuitenkin väiteltävissä, sillä pelin sisällä se on hyvin hienovarainen. Pelaaja saattaa huomaamattaan tiedostaa sen, mutta silti yhteys kuolemaan voi olla vaikea huomata.

Kameran näkymän koon muuttaminen on funktionaalisesti sama kuin oikeassa elämässä kameran zoomaaminen, mutta koska siihen ei liity muita kameran ominaisuuksia, kuten fokusta, on kameran näkymän koon muuttaminen tarkempi termi efektille. Pelkästään kameran ortografinen pinta-ala muuttuu zoomatessa. Työssä kuitenkin käytetään termiä zoomaus näkymän koon muuttamiselle keskenään vaihdettavasti.

Tehokeinona zoomaus on myös varattu lähestulkoon samoille toiminnoille kuin kääntäminenkin. Zoomi toimii myös nopeana liikkeenä, kamera zoomaa sisään hyvin nopeasti ja palaa hitaasti alkuperäiseen näkymän kokoon. Tälle efektille on vaikea löytää oikean elämän vertauskuvaa, joten se toimii abstraktina kameran tehokeinona rotaation kanssa. Kuvassa 33 kameraefektit pelihahmon tippuessa kuiluun.



Kuva 33. Pelihahmo on pudonnut kuiluun ja kameraa zoomaa sekä kääntyy hetkellisesti.

Zoomauksella on kuitenkin peliin tärkeä vaikutus, joka on toiminnan kehystäminen (38). Zoomaus keskittyy aina toiminnan kohteeseen, joka vetää siihen huomiota. Kun pelihahmo pitelee sahaa, kamera zoomaa pelihahmoon hieman samalla tavalla kuin kameran tärinä kasvaa sahaa pidellessä. Tämä auttaa muita pelaajia kiinnittämään huomiota pelaajaan, joka on valmiina vahingoittamaan muita pelihahmoja. Samoin sahaa heitettäessä tai siihen syöksyessä kamera zoomaa kyseisen toiminnan keskipisteeseen tuoden tapahtuman kaikkien pelaajien huomioon hieman paremmin.

Pelihahmon kuollessa kamera zoomaa myös sisään, mutta toisin kuin muissa tapauksissa, kyseessä ei ole niinkään huomion tuonti kyseiseen tapahtumaan, vaan sitä käytetään enemmänkin vain abstraktina tehokeinona, joka lisää tapahtuman tärkeyden tunnetta. Koska kameran näkymän koko on pienempi, efektit kuten tärinä ja työntö korostuvat verrattain enemmän.

Näiden neljän efektin ansiosta Chucksawin kamera on hyvin dynaaminen ja pystyisi jo itsessään antamaan painoa ja tunnetta melkein kaikille pelin tapahtumille, etenkin törmäyksille, ilman muita viimeistelyefektejä. Koska Chucksawissa on aina vähintään kaksi pelaajaa ja peli on hyvin nopeatempoinen, on kamera melkein koko ajan jonkinlaisessa liikkeessä, luoden vaikuttavan illuusion kameran fyysisestä olemassaolosta pelimaailman sisällä.

#### 7.4 Luettavuuden säilyttäminen

Chucksaw on täynnä kaikenlaisia efektejä, ja pelatessa on harvoin hetkiä, jolloin efektejä ei olisi merkittävä määrä ruudulla samanaikaisesti. Tämän takia on tärkeää päättää, mitä tapahtumia oikeasti juissataan ja mitkä tapahtumat jätetään ilman suurempia efektejä. Ensimmäisenä prioriteettina ovat kaikki pelaajan verbit. Sitten tulee sahan interaktiot pelaajan kanssa, jotka ovat pelin päämekaniikat. Kaikki efektit tämän jälkeen pitää puntaroida tarkemmin. Taustan grafiikat ja koristelu on minimaalista, sillä yleisesti ottaen pelihahmot ja kaikki mahdolliset peliobjektit, tässä tapauksessa saha, ovat pääosassa. (31) Peliä ei voi täyttää liian suurella määrällä efektejä, sillä josain vaiheessa aistit ylikuormittuvat ja tärkeä info hukkuu turhien efektien sekaan. Viimeistelyn tulisi tukea pelin muita aspekteja, eikä toisinpäin (13).

Vaikka Chucksawissa on paljon visuaalisia efektejä, ne on suunniteltu olemaan mahdollisimman informatiivisia. Efektit ovat nopeita, joten ne eivät ikinä peitä peliruutua tarpeeksi kauaa, että pelaaja ei näkisi, mitä pelissä tapahtuu. Efektit on myös usein suunniteltu niin, että vaikka ne voivat peittää pelihahmon kokonaan, ne keskittyvät pelihahmon sijaintiin ja seuraavat pelaajaa koko efektin ajan. Vaikka pelaaja ei näekään pelihahmoa, hän silti tietää, missä pelihahmo kulkee, koska hän voi sen sijaan seurata efektiä. Sama sääntö pätee sahan efekteihin.

Efekteillä ei kuitenkaan tarvitse aina olla pelin toiminnan kannalta oleellista infoa. Monet sahan efekteistä ovat olemassa vain muokkaamassa pelaajan käsitystä sahasta ja vakuuttamassa sen fyysikaalisuudesta. Vaikka ne eivät anna merkityksellistä tietoa pelin tilasta, ne auttavat pelaajaa kuvittelemaan pelimaailman oikeana tilana, jossa he ovat pelihahmon kautta. On kuitenkin tärkeää muistaa, että kaikkien efektien tulisi vahvistaa yhtenäistä metaforaa peliobjektista, johon ne on liitetty (12). Tämän lisäksi viimeistely, joka ei ole informatiivista, ei saisi peittää efektejä, jotka välittävät tietoa pelaajalle. Jokainen efekti, joka on liitetty esimerkiksi sahaan, vahvistaa sen metallisuuden, painon ja nopeuden tunnetta. Samoin kaikki pelaajan efektit vahvistavat tunnetta ketterästä, kevyestä entiteetistä, jolla on kuitenkin paljon iskuvoimaa. Jotkut efektit pelissä kuitenkin ovat hieman rajatapauksia, sillä ne voivat olla olemassa lähinnä antamassa informaatiota, ja vaikutelman korostaminen on efektille sekundäärinen tarkoitus. Näissä tapauksissa efektejä on yritetty sovittaa halutun attribuutin vahvistamiseen.

Monet Chucksawin efektit ovat myös puhtaasti tyylivalintoja, jotka antavat sille omaperäisen ulkonäön. Voitaisiin argumentoida, että pelin pikselöitynyt ulkonäkö on täysin tarpeeton, sillä se obfuskoii visuaaleja hyödyttämättä peliä suoranaisesti. Tietyllä tavalla se on totta, sillä korkeampi pikselimäärä antaisi pelihahmoille ja ympäristölle marginaalisesti tarkemmat ulkorajat ja auttaisi

paikkatietoisuuden suhteen. Tämä tyylivalinta kuitenkin mahdollistaa pelin ulkonäön helpomman yhtenäistämisen. Koska tarkemmat yksityiskohdat häviävät pikseleihin, voidaan keskittyä enemmän peliohjainten muotokieleeseen ja väriin. Kehitysprosessi nopeutuu ja grafiikan tuottamisen kompleksisuus pienenee.

## 8 Pohdinta

Pelituntuma määriteltiin ja tunnistettiin kaksi pelituntuman alaosaa, juissi ja öljy. Tämän lisäksi määriteltiin pelituntuman metriikoita, joiden kautta voidaan tarkastella eri pelien pelituntumaa. Opinnäytetyö käytti teoriapohjana hyvin paljon Steve Swinkin kirjaa pelituntumasta, joka on kuitenkin vain yksi näkökulma pelituntumaan. Etenkin Swinkin määritelmä todellisesta pelituntumasta rajaa sen erittäin pieneen valikoimaan peligenrejä ja kaksi eri henkilöä voisivat saapua eri tuloksiin määritellesään, onko pelissä reaaliaikaista kontrollia vai ei. Pelituntuman määritelmä on siis edelleen häilyvä. Kirja nostaa kysymyksen erilaisista pelituntumista, jotka eivät sovi tähän tarkkaan määritelmään. Näille peleille, jotka eivät sovi tähän pelituntuman määritelmään, voitaisiin kehittää muita pelituntuman malleja. Swinkin määrittely kuitenkin helpottaa hahmottamaan useimpien toimintapeliin pelituntumaa.

Kirjallisuuden vähyden vuoksi pelituntumasta ei ole olemassa muita kattavia teorioita ja loppujen lopuksi, vaikka pelituntumaa voidaan yrittää parantaa tietyillä tehokeinoilla, on pelituntuma subjektiivista. Pelituntuman teorian vähyys ehdottaa pelituntuman olevan enemmän taidetta kuin tiedettä. Vaikka pelituntuma on hyvin tärkeä osa videopelin kokonaisuutta, pelituntuman parantamiselle ei ole kehitetty selvää kaavaa tai työnkulkua. Peligenrestä riippuen voidaan kuitenkin erotella yksittäisiä keinoja tai suunnitteluvalintoja, jotka ovat läsnä lähestulkoon jokaisessa genren pelissä. Voitaisiin siis argumentoida näiden olevan integraalinen osa kyseisille peleille. Etenkin tasohyppelypeleissä pelihahmon syötepuskurointi on hyvin laajasti käytössä ja voitaisiin jopa väittää pelituntuman objektiivisesti kärsivän, mikäli syötepuskurointi poistettaisiin tämän genren peleistä.

Pelituntuman metriikat havaittiin kuitenkin erittäin hyödyllisiksi pelimekaniikkoja ja -objekteja tarkastellessa. Vaikka suurin osa metriikoista on subjektiivisia, eikä voida mitata kuinka ”hyvä” esimerkiksi viimeistely on, antavat ne hyödyllisen työkalun vertailuun. Pelihahmon ulkonäköä suunnitellessa metriikoita käyttämällä voitiin havaita, kuinka hyvin suunniteltu metafora ja nykyinen toteutus täsmäsivät toisiaan. Etenkin OCDAssa syötteen ja vasteen suhteen ymmärtäminen auttoi suunnittelemaan mekaniikkoja ja pelihahmon attribuutteja, jotka linjautuivat pelin suunnittelupilarien mukaisesti. Kontekstin avulla tasosuunnittelun iterointi onnistui helpommin ja mahdollisia ongelmia voitiin havaita jo ennen testausta analysoimalla tehtyä tasoa kontekstin kautta.

Riippumatta siitä, sopsisiko mikä tahansa peli Swinkin pelituntuman määritelmään, jokainen peli hyötynee pelituntuman metriikoiden käyttämisestä peliä suunnitellessa. Halutun kaltaisen pelituntuman saavuttamista voidaan helpottaa tarkastelemalla olemassa olevia pelejä pelituntuman metriikoiden kautta ja soveltamalla niistä saatua dataa omaan peliprojektiin. Riippuen siitä, miten peli sopii pelituntuman määritelmään, metriikoiden tärkeys muuttunee. Tasohyppelypelissä on hyvin tärkeää tarkastella syötettä, vastetta ja kontekstia, kun taas taktisessa pelissä tärkeintä lie-  
nee säännöt ja mahdollisimman eheä metafora. Peliprojektissa voitaisiin antaa jokaiselle metriikalle tärkeysaste ja sen mukaan käyttää resursseja pelin alueiden iteroimiseen.

Juissin ja öljyn määrittelyn todettiin kontekstualisoivan viimeistelyä ja vastetta respektiivisesti. Kun juissin tarkoitus ja yleiset implementoimistavat oli ymmärretty, pystyttiin pelihahmon verbejä juissaamaan tarkoituksellisemmin. Juissin yleispiirteitä hyödyntämällä pystyttiin määrittelemään, mitkä tapahtumat oikeasti tarvitsivat näitä efektejä ja millaisia tuntoaistimuksia juissin pitäisi herättää kussakin tilanteessa. Öljyn ymmärtäminen käsitteenä antoi konkreettisemmän ymmärryksen vasteesta metriikkana. Kun pelien suunnittelupilarit ja mekaniikat ikään kuin lajiteltiin tärkeysjärjestykseen, saatiin idea pelihahmojen verbeistä, jotka tarvitsivat öljyämistä. Kun pelimekaniikassa ilmeni ongelmia, voitiin sitä lähestyä öljyn kautta, kyseenalaistaen mekaniikan todellisen tarkoituksen ja öljyämällä helpottaa pelaajaa keskittymään mekaniikan olennaiseen osaan.

On vaikea arvioida opinnäytetyötä varten tehtyjen projektien pelituntumaa puolueettomasti. Metriikoita ja pelitestaajien palautetta hyödyntämällä uskallettaisiin kuitenkin väittää molempien pelien saavuttaneen pelaajille miellyttävän pelituntuman. Molemmat pelit sopivat Swinkin määritelmään todellisesta pelituntumasta, ja siten ne hyötyivät paljon kyseisestä pelituntuman teoriasta.

Suuri osa projektien työmäärästä kului pelihahmojen ohjaamisen tuntuman parantamiseen. Työn tuloksena molempiin peleihin saatiin kuitenkin aikaan pelihahmo, jonka ohjaaminen vastaa hyvin pelaajan odotuksia ja käsittelee syötettä mahdollisimman paljon mukautuakseen pelaajan toiveisiin. Oikeanlaisen vasteen suunnittelussa ongelmia tuotti eniten juuri pelaajan syötteiden oikean tarkoituksen ennustaminen vasteessa. Etenkin rajan vetäminen siihen, kuinka aggressiivisesti peli moduloi pelaajan syötettä, oli hankala määrittää, sillä pelaamista haluttiin helpottaa mahdollisimman paljon, ilman että pelin pelaaminen ei trivialisoituisi. Joskus tuntui, että pahimmillaan öljyäminen saattoi pysäyttää pelihahmon reaaliaikaisen kontrollin. Tehokeinoihin jouduttiin myös ohjelmoimaan paljon rajatapauksia, jotka eivät välttämättä skaalautuisi hyvin suurempiin pelipro-



jekteihin, mutta olivat kustannustehokkaita opinnäytetyötä ajatellen. Tulevaisuudessa näistä tulisi kehittää universaalimpia ratkaisuja, joita ei täytyisi juurikaan muokata siirtyessä uuteen projektiin.

Tasosuunnittelun ja kontekstin välinen suhde osoittautui myös hyvin tärkeäksi molemmissa projekteissa. Kun pelihahmojen verbit kirjoitettiin ylös, voitiin jokaiselle verbille listata käyttötapoja tasoissa ja sen lisäksi listata, miten jokainen verbi ja sen käyttäminen luultavimmin vaikuttaisi tason kontekstiin. Sama prosessi toistettiin tasoihin sijoitettaviin peliobjekteihin. Tätä käyttämällä voitiin helpommin suunnitella tasoja, jotka tuntuivat pelaajasta hyvin erilaisilta, vaikka pelihahmojen verbien ja peliobjektien määrä oli hyvin pieni. Ilman kontekstin hyödyntämistä tasosuunnittelu olisi todennäköisesti riippunut paljon enemmän suuresta määrästä testausta ja erilaisilta tuntuvien tasojen suunnittelu olisi ollut enemmän yrittystä ja erehdystä.

Pelituntumaa oli hankala osoittaa hyväksi, ja pelitestauksessaan pelaajat eivät yleensä pystyneet toteamaan mitään konkreettista pelituntumasta. Testauksessa usein todettiin pelituntuman olevan tarpeeksi hyvä, mikäli pelaajalla ei ollut valitettavaa testattavasta asiasta. Jos esimerkiksi pelin kamera ei tuottanut ongelmia pelaajalle, todettiin sen tuntuman olevan tarpeeksi hyvä OCDAssa. Hyvää pelituntumaa on siis hankala todistaa, mutta huonon pelituntuman pelaaja pystyy havaitsemaan. Pelaajat pystyivät kommentoimaan jonkin asian olevan pielessä, mutta eivät pystyneet sanomaan mitään tiettyä. Vasta videokuvaa pelaamisesta tarkastellessa voitiin alkaa analysoida tarkemmin pelituntuman laatua.

## 9 Yhteenveto

Pelituntuma voidaan jakaa kolmeen eri osa-alueeseen: reaaliaikaiseen kontrolliin, spatiaaliseen simulaatioon ja viimeistelyyn. Jotta pelaajalla voi olla reaaliaikainen kontrolli, hänen täytyy ohjata peliobjektia, yleensä pelihahmoa, pelin sisällä. Pelin tulisi tuottaa vastetta pelaajan syötteisiin vähintään 100ms välein ihmisen vastesyklin mukaisesti, jotta tämä kontrolli ei särkyisi. Spatiaalisella simulaatiolla tarkoitetaan virtuaalista tilaa, jonka kanssa pelihahmo voi olla vuorovaikutuksessa vapaasti. Tila ei voi olla abstrakti eli pelihahmon liikettä ei voida rajoittaa esimerkiksi ruudukkoon. Pelaajan tulisi myös pystyä havaitsemaan tilaa aktiivisesti sen sijaan, että se esiteltäisiin pelaajalle passiivisesti ilman pelaajan interaktiota. Viimeistelyllä tarkoitetaan mitä tahansa pelin osaa, joka ei vaikuta peliin mekaanisella tasolla. Esimerkiksi kaikki grafiikat ja ääniefektit ovat viimeistelyä. Viimeistelyllä kerrotaan lisää peliobjektien luonteesta ja annetaan vaikutelma peliobjektien fyysisyydestä pelimaailmassa. Tämä pelituntuman määritelmä pätee lähinnä toimintapeleihin.

Pelituntumaa voidaan yrittää mitata lähinnä subjektiivisilla metriikoilla. Pelituntuman metriikat ovat syöte, vaste, konteksti, metafora, viimeistely, säännöt ja yhtenäisyys. Pelituntumaa voidaan mitata vertaamalla metriikoita kahden videopelin välillä tai työn nykyistä versiota haluttuun lopputulokseen.

Syöte määrittää pelaajan fyysisen laitteen, jolla hän antaa komentoja pelille. Laitteen painikkeiden laatu, sijoitus ja määrä määrittävät laitteen syötemahdollisuuden syvyyden. Pelaajan käsien fyysinen liike eli tapa, jolla hän antaa syötteen pelille vaikuttaa myös syötteen tuntumaan. Pelaajan syötettä voidaan tallentaa pelaamisen aikana, mitä voidaan tarkastella objektiivisena metriikkana. Vaste on pelin tapa ottaa pelaajan syöte ja käsitellä se liikkeeksi ruudulle. Peli voi ottaa erittäin yksinkertaisen syötteen, kuten painikkeen painalluksen, ja muuntaa sen monimuotoiseksi vasteeksi pelin muuttujien moduloimiseen. Pelaajan syötemahdollisuudet ja pelin vaste määrittelevät yhdessä, kuinka herkkä pelin vaste on syötteelle.

Konteksti mittaa spatiaalista simulaatiota ja sen vaikutusta pelaajaan. Peliobjektien paikoitus ja etäisyys toisistaan vaikuttavat pelituntumaan, samoin kuin tapa, jolla pelihahmo on vuorovaikutuksessa tilan kanssa. Pelihahmon ympäristö määrittää pelihahmolle saatavilla olevat verbit.

Metafora määrittää, kuinka helposti pelaaja voi arvioida peliobjektin tarkoituksen pelimaailmassa sen ulkonäön perusteella. Tyyllittelyn laatu määrittää pelaajan odotukset peliobjektista. Realistisen näköisen peliobjektin tulisi käyttäytyä kuin oikeassa elämässä, kun taas abstraktimpi ulkonäkö ei herätä yhtä yksityiskohtaisia odotuksia. Viimeistely määrittää peliobjektin fyysisen luonteen.

Peliobjektin grafiikka ja äänet muodostavat yhdessä eheän kuvan objektin esittämästä asiasta. Tarkastelemalla viimeistelyä pelaaja voi yrittää arvioida peliobjektin mahdollista painoa tai tekstuuria.

Säännöillä tarkoitetaan pelin eri muuttujien suhteita toisiinsa. Nämä suhteet itsessään antavat muuttujille ja peliobjekteille merkityksen pelin sisällä. Kun säännöt sidotaan esimerkiksi pelissä edistymiseen, pelaajalle voidaan luoda pidempiaikainen motivaatio olla vuorovaikutuksessa pelimekaniikkojen kanssa.

Yhtenäisyydellä mitataan kuinka hyvin eri metriikat sopivat toisiinsa. Viimeistelyn ja vasteen tulisi vahvistaa yhtenäistä metaforaa peliobjekteista ja -ympäristöstä. Sääntöjen ja kontekstin tulisi motivoida pelaajaa pelaamaan mahdollisimman miellyttävällä tavalla hyödyntäen useita pelimekaniikkoja. Pelin tulisi myös hyödyntää pelaajan käyttämää syötelaitetta mahdollisimman hyvin.

Juissilla tarkoitetaan nopeita viimeistelyefektejä, joilla rikastetaan pelin antamaa palautetta pelaajan syötteeseen. Juissia voidaan käyttää myös peliobjektien tilan vaihtumisen yhteydessä tai uuden peliobjektin syntyessä. Efektejä kasaamalla voidaan vaikuttaa lähinnä pelin metaforaan ja viimeistelyyn. Hyvin juissattu pelihahmon verbi on itsessään tyydyttävä suorittaa. Pelin öljyminen taas helpottaa itse syötteen antamista. Pelaajalle annetaan suurempi aikajana, jonka sisällä ilmaista haluamansa vaste pelin sisällä. Hyvin öljytty peli yrittää päätellä pelaajan syötteiden tarkoituksen joustavasti ja tuottaa pelaajan haluaman vasteen olosuhteissa, joissa se ei olisi muuten mahdollista. Öljymisen tuloksena pelaajan syöte tuntuu johdonmukaisemmalta.

## Lähteet

1. **S, Swink.** *Game feel : a game designer's guide to virtual sensation.* Burlington (MA) : Morgan Kaufmann, 2009. 978-0-12-374328-2.
2. **J, Nijman.** Art Of Screenshake. *YouTube.* 16. 12 2013. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=AjEqssNZ-U>.
3. **Jonasson M, Purho P.** Juice it or lose it. *YouTube.* 2012. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=Fy0aCDmgnxg>.
4. **N, Berbece.** Game Feel: Why Your Death Animation Sucks. *GDCVault.* 2015. [Viitattu: 3. 11 2020.] <https://www.gdcvault.com/play/1022759/Game-Feel-Why-Your-Death>.
5. **Z, Gage.** Controls You Can Feel: Putting Tactility Back Into Touch Controls. *YouTube.* 2012. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=OIOitNx9RHI>.
6. **M, West.** Measuring Responsiveness in Video Games. *Cowboy Programming.* 2008. [Viitattu: 3. 11 2020.] <http://cowboyprogramming.com/2008/05/30/measuring-responsiveness-in-video-games/>.
7. **TheGamingBritShow.** Donkey Kong Jungle Beat Review. *YouTube.* 28. 9 2020. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=c0Kalcys24U>.
8. **D, Taylor.** Ten Principles For Good Level Design. *YouTube.* 2013. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=iNEe3KhMvXM>.
9. **A, Myers.** Let's expand the meaning of "GAME FEEL"!! *YouTube.* 2019. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=qaCBpgVlw0Q>.
10. **F, Kelly.** Don't Juice It or Lose It. *GDCVault.* 2014. [Viitattu: 3. 11 2020.] <https://www.gdcvault.com/play/1020861/Don-t-Juice-It-or>.
11. **D, Orosz.** Shaping Emotions: Utilizing Shape Language and Symbols in Level Design. *Do Game Design.* [Viitattu: 3. 11 2020.] <https://dogamedesign.com/shaping-emotions>.
12. **L, Brown.** The Nuance of Juice. *YouTube.* 2016. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=qtgWBUIOjK4>.
13. **Gray K, Gabler K, Shodhan S, Kucic M.** How to prototype a Game in Under 7 Days. *Gamasutra.* 2005. [Viitattu: 3. 11 2020.] [https://www.gamasutra.com/view/feature/130848/how\\_to\\_prototype\\_a\\_game\\_in\\_under\\_7\\_.php](https://www.gamasutra.com/view/feature/130848/how_to_prototype_a_game_in_under_7_.php).
14. **Juutas.** I Like Hitstop In Video Games. *YouTube.* 2018. [Viitattu: 3. 11 2020.] [https://www.youtube.com/watch?v=\\_jFbgoHyZnM](https://www.youtube.com/watch?v=_jFbgoHyZnM).
15. **D, Daniels.** Why some games feel better than others - part 3. 2013. [Viitattu: 3. 11 2020.] <http://derekdaniels.com/blog/2013/2/27/game-feel-part-3>.
16. —. Helping the player out. 2013. [Viitattu: 3. 11 2020.] <http://derekdaniels.com/blog/2013/7/23/helping-the-player-out>.

17. **L, Doucet.** Oil it or Spoil It! *Gamasutra*. 2016. [Viitattu: 3. 11 2020.]  
[https://www.gamasutra.com/blogs/LarsDoucet/20160810/279009/Oil\\_it\\_or\\_Spoil\\_it.php](https://www.gamasutra.com/blogs/LarsDoucet/20160810/279009/Oil_it_or_Spoil_it.php).
18. **S, Benard.** Dead Cells post-mortem. *YouTube*. 2018. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=YKWNHhh3WAw>.
19. —. How Dead Cells Secretly Stops You From Dying. *GameSpot*. 2019. [Viitattu: 3. 11 2020.]  
<https://www.gamespot.com/videos/how-dead-cells-secretly-stops-you-from-dying/2300-6449363/>.
20. **S, Coster.** Forgiveness Mechanics: Reading Minds for Responsive Gameplay. *YouTube*. 2020. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=HCnZhs-92j0>.
21. **M, Brown.** Why Does Celeste Feel So Good to Play? | Game Maker's Toolkit. *YouTube*. 31. 7 2019. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=yorTG9at90g>.
22. **P, Wilson.** Opinion: Analog Input Processing And Delivering Great Controls . *Gamasutra*. 17. 6 2011. [Viitattu: 5. 11 2020.]  
[https://www.gamasutra.com/view/news/125467/Opinion\\_Analog\\_Input\\_Processing\\_And\\_Delivering\\_Great\\_Controls.php](https://www.gamasutra.com/view/news/125467/Opinion_Analog_Input_Processing_And_Delivering_Great_Controls.php).
23. **M, Brown.** Who Gets to be Awesome in Games? *YouTube*. 2020. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=XSvclSkmdyY>.
24. **M, Worch.** Meaningful Choice in Game Level Design. *YouTube*. 2014. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=BEF4GVNzkUw>.
25. **A, Saltsman.** Tuning Canabalt. *Gamasutra*. 29. 9 10. [Viitattu: 8. 11 2020.]  
[https://www.gamasutra.com/blogs/AdamSaltsman/20100929/88155/Tuning\\_Canabalt.php](https://www.gamasutra.com/blogs/AdamSaltsman/20100929/88155/Tuning_Canabalt.php).
26. **Gage Z, Nealen A, Suttner N, Wilson D.** Eggplant: Secret Lives of Games. *Maximizing Minimalism with Disc Room*. 2020. [Viitattu: 3. 11 2020.] <https://eggplant.show/55-maximizing-minimalism-with-disc-room>.
27. **J, Anderson.** The Witcher 2 - A Grand Experiment. *YouTube*. 2020. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=htYR2GdA7OE>.
28. **K, Pittman.** Math for Game Programmers: Building a Better Jump. *YouTube*. 2016. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=hG9SzQxaCm8>.
29. **D, Daniels.** Why some games feel better than others - part 2. 2013. [Viitattu: 3. 11 2020.]  
<http://derekdaniels.com/blog/2013/2/23/game-feel-part-2>.
30. **I, Thomas.** Planning – The Core Reason Why Gameplay Feels Good. *Frictional Games*, 30. 5 2017. [Viitattu: 8. 11 2020.] <https://frictionalgames.com/2017-05-planning-the-core-reason-why-gameplay-feels-good/>.
31. **C, Portman.** Video Game Readability. 2018. [Viitattu: 3. 11 2020.]  
<https://www.caseportman.com/single-post/2018/01/24/Video-Game-Readability>.
32. **D, Root.** Fluid Motion | Celeste's Animation. *YouTube*. 2018. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=h2VMnzawfh0>.

33. **J, Sutphin.** Doing Thumbstick Dead Zones Right. *Gamasutra*. 16. 4 2013. [Viitattu: 5. 11 2020.]  
[https://www.gamasutra.com/blogs/JoshSutphin/20130416/190541/Doing\\_Thumbstick\\_Dead\\_Zones\\_Right.php](https://www.gamasutra.com/blogs/JoshSutphin/20130416/190541/Doing_Thumbstick_Dead_Zones_Right.php).
34. **M, Brown.** The Secret of Mario's Jump (and other Versatile Verbs) | Game Maker's Toolkit. *YouTube*. 2017. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=7daTGyVZ60I>.
35. **A, Dazé-Hill.** Juicy Beast. 12. 3 2014. [Viitattu: 8. 11 2020.]  
<http://juicybeast.com/2014/03/12/the-making-of-toto-temple-deluxe-the-unique-experience/>.
36. **Steg L, Younas O.** Representation of UI. *Game UX Master Guide*. 2019. [Viitattu: 3. 11 2020.]  
[https://gameuxmasterguide.com/2019-04-24-UI\\_Representation/](https://gameuxmasterguide.com/2019-04-24-UI_Representation/).
37. **I, Keren.** How Cameras in Side-Scrollers Work. *YouTube*. 2015. [Viitattu: 3. 11 2020.]  
<https://www.youtube.com/watch?v=pdvCO97jOQk>.
38. **S, Eiserloh.** Math for Game Programmers: Juicing Your Cameras With Math. *YouTube*. 2016. [Viitattu: 3. 11 2020.] <https://www.youtube.com/watch?v=tu-Qe66AvtY>.