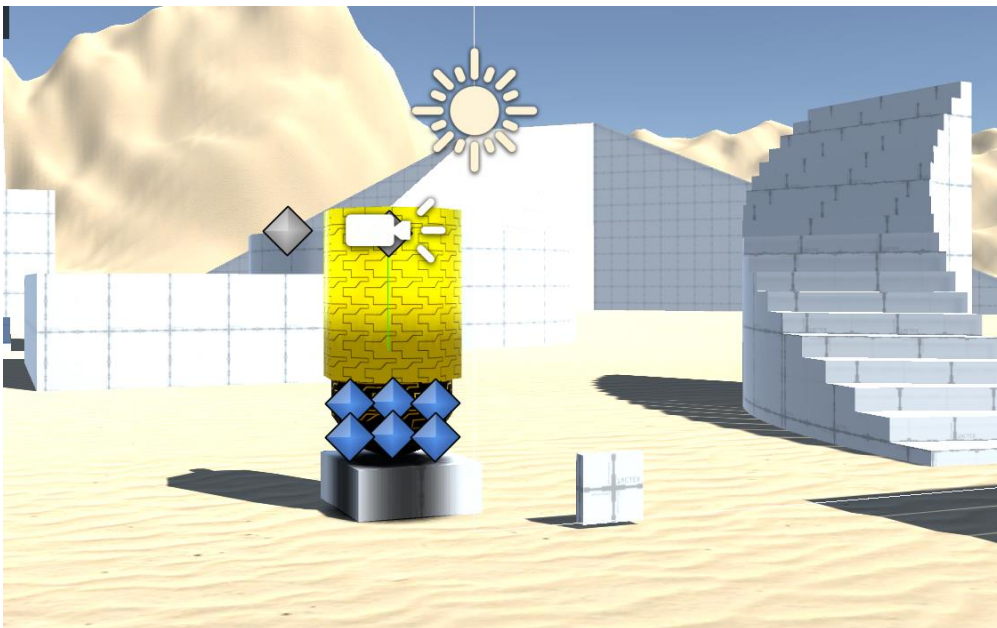


Aleksi Nieminen

3D-tasoloikkapelin liikkumisen toteuttaminen



Tradenomi (AMK)
Tietojenkäsittely koulutus
Kevät 2023



KAMK • University
of Applied Sciences

Tiivistelmä

Tekijä(t): Aleksi Nieminen

Työn nimi: 3D-tasoloikkapelin liikkumisen toteuttaminen

Tutkintonimike: Tradenomi (AMK)

Asiasanat: Unity, 3D-tasoloikka

Opinnäytteen aiheena oli käsitellä 3D-tasoloikkapelin liikkumisen mekaniikkojen toteuttaminen, erityisesti erikoisliikkeiden toteuttaminen. Opinnäyte perustui peliprojektiin, jonka tekeminen alkoi 2019. Tämän projektin tarkoituksena oli toimia tekijälleen harjoituksena, miten käyttää Unity-pelimoottoria paremmin. Opinnäytteen tavoitteena oli näyttää, miten tiettyjä erikoisliikkeitä voidaan tehdä ja mitä asioita on hyvä ottaa huomioon, kun tekee liikkumismekaniikkoja peliin.

Teoriaosuudessa käytiin erilaiset kamerakulmat, peligenret ja niiden historiaa. Peligenre antaa vain suuntaa antavaa tietoa, koska jotkin mekaniikoista eivät ole sidottuna vain tiettyyn genreen, mutta genre voi rajata sitä. Kamerakulmat ja niiden historiaa käytiin läpi. Samoin käsiteltiin pelisarjoja ja mekaniikkoja, joista otettiin inspiraatiota. Myös kerrottiin, miten Unity-pelimoottorin komponentit toimivat ja millaisia rajoituksia niissä on. Käsiteltiin testausta ja miksi se pitää tehdä mahdollisimman pian, kun projekti alkaa olla vähänkin pelattava.

Toteutuksessa käytiin läpi, mitä tehtiin projektiin. Ensimmäisenä käytiin, mistä peleistä otettiin inspiraatiota ja mitä otettiin inspiraatioksi näistä peleistä. Projektissa käytettiin CharacterController-komponentin sijaan Rigidbody-komponenttia. Syy oli ettei, CharacterController-komponenttia ei voinut kääntää kuin Y-akselilla, mutta Rigidbody-komponenttia voidaan kääntää. Hyppy-, tuplahyppy- ja coyote time -mekaniikat ovat olennainen osa tätä peliä. Pelaajahahmo pystyy tarttumaan kielekkeen reunasta. Projekti jäi kesken-eräiseksi, mutta sen on tarkoitus joskus valmistua valmiiksi peliksi. Projektin tekijä onnistui siinä, että oppi käyttämään Unitya paremmin.

Että peliin saadaan erikoisliikkeitä, pelissä pitää olla toimiva perusliikkuminen. Monet erikoisliikkeet tässä projektissa olivat riippuvaisia perusliikkeistä, jotka ovat kävely ja hyppiminen. Pelimoottorin päivittyessä kannattaa seurata, onko siihen tullut mekaniikka valmiiksi tehtynä pelimoottoriin.

Abstract

Author(s): Aleksi Nieminen

Title of the Publication: Making Movement Mechanics for 3D Platformer Game

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: Unity, 3D Platformer

The subject of the thesis is to go through with the implementation of movement mechanics in a 3D platformer game, especially the implementation of special moves. The thesis was based on a game project that started in 2019. The purpose of this project was to be an exercise for its maker on how to use the Unity game engine better. The purpose of the thesis is to show how certain special moves can be made and what things should be considered when making movement mechanics for the game.

The theory part covered different camera angles, game genres and history of these. Game genre only section provides indicative information, because some of the mechanics are not tied to any certain genre, but the genre can limit it. Camera angles and their history was a part of this thesis. The inspiration from certain game series was discussed. Along, with the components of the Unity game engine and their limitations. Testing and why it must be done as soon as possible, when the project is even a little bit playable has also been a subject of this thesis.

In the implementation, to stages of the project were reviewed. The first stage was to select the games for inspiration and discuss what was inspirational in these games. Instead of the CharacterController component, the Rigidbody component was used in the project. The reason was that the CharacterController component could only be rotated both on Y-axis, but the Rigidbody component could be rotated. The jump, double jump and coyote time mechanics are an important part of this game. The player character can grab the edge of the ledge. The project was left unfinished, but it is supposed to be finished as a full game someday. The creator of the project succeeded in learning to use Unity better.

In order to get special moves in the game, the game must have functional basic movements. Many of the special moves in this project depended on the basic moves, which are walking and jumping. When the game engine is updated, it is worth keeping an eye on whether the mechanics have been made in game engine.

Sisällys

1	Johdanto	1
2	Pelaajan hahmon liikkuminen peleissä.....	2
2.1	Ensimmäisen persoonan pelit	2
2.2	Kolmannen persoonan pelit	3
2.3	Roolipelit	5
2.4	Tasohyppelypelit	6
2.5.1	Sly Cooper -pelisarjat	7
2.5.2	Ratchet and Clank -pelisarjat	8
2.5.3	Jak and Daxter -pelisarjat	9
2.5.4	Mekaniikoista	10
3	Unity-pelimoottori.....	13
3.1	CharacterController-komponentti	13
3.2	Rigidbody-komponentti	13
3.3	Input Manager	14
3.4	Input System.....	14
3.5	Spline-komponentti.....	14
4	Pelitestauksesta	15
5	Toteutus.....	16
5.1	Pelaajan hahmon liikkuminen	17
5.2	Liikkumistyylin muuttuminen ja taistelussa liikkuminen	19
5.3	Ympäristöstä riippuvaiset mekaniikat	20
5.4	Projektin jatkokehitys.....	23
6	Pohdinta	25
7	Yhteenveto	26
	Lähteet	27

Symboliluettelo

FPS	First-person shooter eli ensimmäisen persoonan ammuntopeli.
KeyCode	Unityssä, ottaa sisääntulon suoraan näppäimistöä tai ohjaimesta.
Raycast	Suora viiva tiettyyn suuntaan ja koodi katsoo, jos se osuu johonkin.
Rigidbody	Unityssa, fysiikkasimulaatio pystyy vaikuttamaan objektin sijaintiin
Solmu	Solmut on linkitetty toisiinsa muodostaen verkkoja ja sisältävät tietoa.
Spherecast	Raycast, mutta pallo menee pitkin ja ilmoittaa, jos jotain osuu palloon.
Spline	Käyriä, joilla on solmuja, nämä solmut määrittävät näiden kaaren.
Unity 3D	3D-pelimoottori.

1 Johdanto

Opinnäytetyön aiheena on käydä läpi, miten pelaajahahmon liikkuminen tehtiin peliprojektiin. Pelaajahahmon liikkumista käydään vähemmän, kun taas erikoisliikkuminen on pääaiheena. Tietenkin tarvitaan perusliikkumismekaniikkoja, että voidaan käyttää näitä erikoisempia liikkumismekaniikkoja ja osa näistä on johdettu perusliikkumismekaniikoista. Projektin tavoitteena oli tehdä Unity-pelimoottorin avulla 3D-toimintatasoloikkapeli ohjelmoinnin näkökulmasta. Tämä projekti oli alun perin tarkoitus olla vain vapaan ajan projekti, jonka tarkoituksena pääasiassa olla harjoitus tekijäänsä varten, että oppisi käyttämään Unity-pelimoottoria paremmin. Suurimmat inspiraation lähteet tähän projektiin ovat peräisin Sony PlayStation 2 -peleistä, jotka ovat Sly Cooper, Ratchet and Clank ja Jak and Daxter. Mutta on olemassa muita pelejä, joissa on samanlaisia tai samantapaisia mekaniikkoja.

Pääasiassa mekaniikat, jotka tässä käydään, liittyvät liikkumiseen. Näitä mekaniikkoja ovat kävely, juoksu, hyppiminen ja erikoisliikkumismekaniikat, kuten köydellä kävely. Pelissä tarvitaan pelaajahahmon tavallisia liikkumiskeinoja, kuten kävelyä, että erikoisliikkumismekaniikat voisivat toimia.

Tässä käydään myös eri peligenrejen mekaniikkoja ja miten ne eroavat tasoloikkapelien mekaniikoista, näitten historiasta on myös hieman mainittu. Käsitellään myös Unity-pelimoottorin eri komponentteja ja niiden toimintaa. Näiden mekaniikkojen mahdollisia toteutustapoja käydään läpi. Pelitestauksesta ja sen tuloksia käsitellään. Tämän opinnäytetyön tarkoituksena on käydä läpi mitä, erilaiset liikkumismekaniikat saattavat tarvita, että ne saadaan toteutettua ja mitä on hyvä ottaa myös huomioon niitä tehdessä.

2 Pelaajan hahmon liikkuminen peleissä

Pelaajan hahmojen liikuttaminen riippuu peligenrestä, toteutuksesta ja minkälainen kamera-kulma pelissä on. Peligenret eivät välttämättä kerro, miten hahmot voivat liikkua. Genre voi kuitenkin rajata sitä, kuten tasoloikkapeleissä liikkuminen on osa genren määritelmää, mutta esimerkiksi pulmapelit voivat olla erilaisia keskenään, kuten EA:n Tetris ja Valven Portal-pelit, joissa liikkuminen on hyvin erilaista. Peligenret ovat siis hieman epämääräisiä ja antavat suuntaa antavaa tietoa. [1.]

2.1 Ensimmäisen persoonan pelit

Ensimmäisen persoonan pelit on kuvattu hahmon näkökulmasta [2]. Näissä peleissä pelaajat eivät näe yleensä kuin kätensä ja mahdollisen aseensa, mutta ei muuta pelattavasta hahmosta. Kaikista suosituin muoto ensimmäinen persoonan peleistä on FPS-pelit [3].

Ensimmäinen FPS-peli oli nimeltä Maze War -niminen peli, se tehtiin NASA:n Imlac PDS-1 -tietokoneelle 1970-luvun alkupuolella. Kyseistä peliä ei julkaistu, koska tämä peli toimi enemmän käytännön toteutuksena kuin myytävänä tuotteena. Pelissä liikuttiin yksi ruutu kerrallaan. Yksi isoimmista edistysaskelista tälle genrelle oli siirtyminen kolikkopeleihin. Se oli Atarin Battlezone, vektorigrafiikalla tehty peli, jossa ohjataan tankkia. [4.] Pelissä on tarkoitus tuhota tankkeja ja lentäviä lautasia kääntämällä pelihahmoa, mutta ylös tai alas katsominen ei ollut mahdollista [5].

Ensimmäinen kotikoneilla pelattava ensimmäisen persoonan ammuntopeli, joka toimi nopeasti verrattuna edellisiin peleihin vuonna 1992, oli ID Softwaren Wolfenstein 3D. Kahdeksantoista kuukauden päästä julkaistiin ID Softwaren Doom, joka oli nopeatempoisempi, kehittyneempi ja siinä oli korkeusvaihtelu toisin kuin Wolfenstein 3D:ssä, mutta ylös tai alas ei pystynyt katsomaan. Raven Softwaren Heretic-peli käytti samaa pelimoottoria kuin Doom, mutta lisäsi heidän peliänsä varten pelimoottoriin ylös ja alas katsomisen. 1996 ID Softwaren Quake julkaistiin, ja tämä peli käytti ensimmäisinä FPS-peleissä täysin 3D-ympäristöjä verrattuna edellisiin peleihin. Kuvassa 1 näkyy Quake-peli, joka on visuaalisesti parannettu versio. Quakessa oli myös rakettihyppymekaniikka. [4.] Rakettihyppy tarkoittaa sitä, kun pelaajaa laukaisee itsensä räjähteiden avulla ilmaan käyttäen sitä liikkumiseen, joko pääsemällä korkeammalle tai nopeammin. Quake ei ollut ensimmäinen peli, jossa oli rakettihyppy, vaan teki siitä tunnetumman [6].



Kuva 1. Quaken grafiikat ovat melko yksinkertaisia, vaikka niitä on paranneltu uudelleen julkaisussa. [7]

2.2 Kolmannen persoonan pelit

Kolmannen persoonan pelit (englanniksi Third person), on kuvattu hahmon ulkopuolelta, jolloin pelaajahahmo näkyy joko kokonaan tai osittain. Tosin kuin ensimmäisen persoonan peleissä kamera voi käyttäytyä eri lailla eri pelissä. Kamera voi olla niin, että se on tietyssä kohtaa pelialuetta, pelaaja ei voi suoraan vaikuttaa kameraan; mutta kamera voi vaihtaa pelialuetta, kun pelaaja siirtyy toiselle alueelle, niin kamera vaihtaa paikkaa. Toinen tapa on, kun kamera seuraa pelaajaa koko ajan, mutta pelaajalla ei ole valtaa tähän, kuten kuvassa 2 olevassa Resident Evil -pelissä.



Kuva 2. Resident Evil HD-versio pelistä. [8]

Kolmas tapa muistuttaa edellistä kameraa, mutta pelaaja voi ohjata kameraa itse eli pelaaja voi kääntää kameraa ja katsoa, mitä hahmon ympärillä tapahtuu. Kolmannen persoonan kamerakulmaa voidaan käyttää monissa peleissä, kuten ammuntapeleissä, mutta silloin se, että pelaajahahmo näkyy voi vaikeuttaa tätä. Tätä tapaa käytetään nykyään eniten näistä tavoista ja taas ensimmäisenä mainittua kameraa ei juurikaan enää käytetä. [3.] Ratchet & Clank -pelisarjassa tämä on ratkaistu kahdella tavalla antamalla pelaajalle automaattisen tähtäyksen (englanniksi auto-aim) ja pistämällä aseet lukittautumaan vihollisia kohti käyttämällä lukittautumistilaa. Lukittautumistilassa pelaajahahmo ja kamera katsovat koko ajan kohdetta kohti. Kuvassa 3 näkyy käytännössä, miten se tapahtuu. [9.]



Kuva 3. Kuvankaappaus Ratchet and Clank 2:sta lukittautumistila päällä.

2.3 Roolipelit

Roolipelit ovat saaneet alkunsa pöytäroolipeleistä ja sieltä ne ovat tulleet tietokoneille. Pöytäroolipeleissä on pelinvetäjä, mutta tietokoneroolipeleissä tätä roolia hoitaa tekoäly. Roolipeleissä yleensä taktikointi on tärkeää ja pelaaja saattaa ohjata useampaa hahmoa. Joissakin roolipeleissä taistelu hoidetaan valikoiden kautta, kun taas toisissa reaaliajassa. Näissä peleissä narratiivinen osuus on tärkeä, mutta hahmon kehittäminenkin on tärkeä, että hahmosta tulee vahvempi ja se saa uusia taitoja, myös ympäristöön vaikuttaminen on tärkeää. [10.] Larian Studion julkaisema Divinity Original Sin 2 -pelissä taistelutilanteessa jokainen hahmo liikkuu ja tekee mitä, tekee omalla vuorollaan [11]. Kuvassa 4 näkyy taistelutilanne Divinity-pelissä.



Kuva 4. Kuvankaappaus Divinity Original Sin 2:n taistelutilanteesta.

Roolipelit, kuten muutkin peligenret, voidaan yhdistää muiden peligenrejen kanssa, joten miten peli toimii voi vaihdella, kuten hahmojen liikkuminen [10]. Esimerkiksi Bethesda Game Studios The Elder Scrolls V: Skyrimissä pelaajaahahmo voi hyppiä koko ajan [12] ja pelaajahahmolla on lohikäärmehuutoja. Yksi huudoista on Whirlwind Sprint, joka mahdollistaa esimerkiksi pudotusten ylittämisen, joka ei hypyllä onnistuisi ja nopean liikkumisen ovelle, jolle on tietty aika päästä ennen kuin se sulkeutuu [13]. Taas Divinity Original Sin 2 -pelissä jatkuva hyppiminen ei ole samalla tavalla mahdollista. Tässä pelissä on kykyjä, jolla hahmot pystyvät hyppäämään, kuten Tactical retreat -taidolla [14]. Tämä ei paljoa mekaanisesti eroa teleporttikyvystä, jotka siirtävät hahmon toiseen kohtaan eroina vain visuaalinen efekti, jälkivaikutus ja kehen sen voi käyttää [15].

2.4 Tasohyppelypelit

Tasohyppelypeleissä hahmon liikuttaminen on tärkeä osa peliä, mihin kuuluu esteiden yli hyppiminen ja väistäminen. Tällaisia esteitä voivat olla rotkot, piikit ja viholliset. Näitä pelejä on sekä 2D- ja 3D-peleinä. Kun on kyse tarkoista hypyistä, joissa pitää laskeutua pienelle alueelle tai hypätä juuri ennen reunaa, pelaajalla saattaa olla vaikeuksia arvioida etäisyyksiä, vaikka peli on kolmiulotteinen. Tämä johtuu siitä, että peliä pelataan näytöltä ja näyttö piirtää näytön tapahtumat kaksiulotteisena. Tämä syvyyden puute voidaan korjata esimerkiksi antamalla pelattavalle hah-

molle blob shadow -varjo, joka on suoraan hahmon alla, vaikka valo tulisi ihan eri suunnasta. Hahmon liikkuminen pitää toimia samalla tavalla joka kerta, kun hahmo esimerkiksi hyppää. Haasteena ei pidä olla hahmon kontrollointi, vaan peli tason navigointi. Tietenkään hahmon liikkuminen ei saa tuntua siltä kuin hahmo kävelisi jäällä paitsi, kun hahmo kävelee jäällä. [16.]

2.5.1 Sly Cooper -pelisarjat

Sucker Punch Productionin Sly Cooper -pelisarjan pelit ovat tasohyppely- ja hiiviskelypelejä [17]. Niissä ohjataan kolmea hahmoa, jotka ovat erilaisia. Pelien Sly-hahmo pystyy juoksemaan ja kiipeämään köysiä, putkia tai vastaavia pitkin, kuvassa 5 näkyy Sly juoksemassa köydellä.



Kuva 5. Kuvankaappaus Sly 3: Konnien kunnia. Sly juoksee köydellä.

Tämä tapahtuu, kun pelihahmo on lähellä kiiltoa ja pelaajaa painaa ympyränäppäintä. Pelihahmo pystyy myös hyppäämään pienille pisteille, jota kutsutaan pelissä tornihypyksi, tämä liike näkyy kuvassa 6.



Kuva 6. Kuvankaappaus Sly 3: Konnien kunnia. Sly ninjatornihypyssä.

Peli etenee tehtävä kerrallaan, ja tehtäviä voi tehdä vapaassa järjestyksessä sitä mukaan, kun niitä tulee. Hahmot voivat vuorovaikuttaa ympäristössä oleviin objekteihin, kuten nappeihin, joka aktivoi esimerkiksi ansaluukun. Pelisarjassa ohjataan myös Bentley- ja Murray-hahmoja, nämä hahmot eivät pysty yhtä ketteriin liikkeisiin, mutta näillä hahmoilla on muita kykyjä. [18.]

2.5.2 Ratchet and Clank -pelisarjat

Ratchet and Clank -pelisarja on seikkailu-, tasohyppely- ja toimintapeli. Pelissä ohjataan pääasiassa Ratchet-hahmoa, hän on ketterä, mutta ei niin ketterä kuin Sly-pelisarjan päähahmo, hahmolla on peleissä iso arsenaali aseita ja apunaan Clank-robotti, jonka avulla hän pystyy liitämään ja hyppäämään korkeammalle. Välillä pelaajan on ohjattava Clank-hahmoa pelkästään. Clankilla pelaaminen on erilaista, koska Clankin pitää komentaa muita robotteja päästääkseen tavoitteeseen. Clankilla on myös jättimuoto, mutta tässä muodossa Clankin pelattavuus muuttuu taistelupainotteiseksi.

Ratchet and Clank -pelisarjasta Ratchet-hahmoa ohjataan pääasiassa. Ratchetin liikkeisiin kuuluvat kameralukitus pelaajahahmon taakse, railgrind-mekaniikka, magneettikengät, rakettikengät,

swingshot-mekaniikka, kielekkeelle tarttuminen ja liikkuminen, seinähypy ja liito. [19.] Kuvassa 7 näkyy, miltä railgrind-mekaniikkaa näyttää.



Kuva 7. Kuvankaappaus Ratchet-hahmo käyttää railgrind-mekaniikkaa

Ratchet and Clank -pelisarjassa Ratchet ja Clank pystyvät liikkumaan sivuttain, kun roikkuvat kielekkeellä ja näin pystyvät pääsemään toisiin paikkoihin. Tätä käytettiin esimerkiksi pelin toisessa osassa Barlow-kentän platinapulttireitillä. [20.]

2.5.3 Jak and Daxter -pelisarjat

Jak and Daxter -pelisarjassa ohjataan Jak- ja Daxter-hahmoja. Pelisarja on tasohyppely- ja toimintapeli. Tässä pelisarjassa ohjataan pääasiassa Jak-hahmoa ja hänen liikeisiinsä kuuluvat muun muassa korkea-, kyykystä-, pitkä- ja kierähdyshyppy ja kierähdysliikkeet. Jakilla on apunaan Daxter, joka on näätämainen pieni hahmo, joka liikkuu ketterästi pitkin pelialuetta, mutta on rajoitettu tiettyihin tilanteisiin ja alueisiin niin kuin Clank. Daxter ei pysty myöskään hyppäämään niin pitkälle tai korkealle kuin Jak ja on muutenkin heikompi hahmo. Pelisarjassa on tankoja, joista Jak voi pyörähtää ympäri ja heittää itsensä joko toiselle tangolle tai alustalle. Hänellä on myös leijulauta, jolla voi mennä nopeammin ja tasapainolla kapeiden putkien tai raiteiden päällä. Näiden

kapeiden putkien päällä ei voi kävellä, koska muuten putoaisi. [21.] Kuvassa 8 näkyy Jakin kierähdysshyppy.



Kuva 8. Kuvankaappaus Jak kierähdysshyppä.

2.5.4 Mekaniikoista

Ponnahduslauta on yleinen mekaniikka tasohyppelypelissä ja se laukaisee pelaajan ylöspäin, kun pelaaja osuu siihen. Ponnahduslauta on ottanut useamman eri muodon, kuten iso sieni, mutta yleensä se on jonkinlainen levy, jonka pohjassa on jousi. [22.]

Leijuva alusta on alusta, joka leijuu ilmassa. Fysiikanlait eivät aina päde niihin, mutta joskus niille on annettu syy, etteivät ne vain lennä ilmassa. Alusta voi olla kiinnitetty niin, että se pysyy ilmassa. Liikkuvat alustat ovat leijuvia alustoja, mutta ne liikkuvat ennalta määritetyllä radalla. [23.]

Kun katoavien alustojen päälle astuu, ne joko katoavat tai tulevat käyttökelvottomiksi ajan kanssa tai niiden päälle astuttaessa. Myös pois hypätessä ne voivat tulla myös hyödyttömiksi tai jopa näiden kombinaatio voi aiheuttaa tämän. Kolmas vaihtoehto on, että nämä alustat katoavat ja ilmestyvät takaisin tietyn aika välein. [24.]

Pohjaton kuoppa on yksi esteistä, jotka saattavat välittömästi tappaa pelaajahahmon tai tehdä vähän vahinkoa. Tämä riippuu pelistä tai jopa pelin tilanteesta joissakin tapauksissa. Pohjaton

kuoppa ei aina näytä pohjattomalta, vaan kuopan pohjalla saattaa olla piikkejä, vettä, laavaa tai jotain muuta, joka voisi tappaa pelaajahahmon. [25.]

Tikkailla kiipeäminen on melko yleinen mekaniikka peleissä, mutta se ei ole yhdenmukaistettu, kuten hyppiminen ja liikkuminen. Tämän takia joissakin peleissä tikkaiden käyttäminen saattaa johtaa pelaajahahmon ennenaikaiseen kuolemaan, kun pelaaja yrittää päästä ylös tai alas tikkaita. Jo miten hahmo aloittaa tikkaille kiipeämisen vaihtelee, joissakin peleissä kiipeämistä varten pitää käyttää käytä-näppäintä tai kun kävelee tikkaita päin pelaajahahmo aloittaa kiipeämisprosessin. Joissakin peleissä kiipeäminen tapahtuu automaattisesti, kun toisissa pelaajan pitää painaa painiketta, että pääsee ylös tai alas päin. Kuvassa 9 näkyy Jak 3 tikkaille kiipeämisen mekaniikka toiminnassa.



Kuva 9. Kuvankaappaus Jak 3 tikkaille kiipeämismekaniikka.

Rockstar Gamesin Grand Theft Auto V -pelissä tikkaille kiipeäminen on vaarallisempaa kuin laskuvarjohyppy tai laitesukellus, koska pelaajat todennäköisemmin putoavat alas, kun yrittävät käyttää tikkaita. Tikkaat ovat myös taipuvaisia rikkomaan pelejä, kuten pelaaja ei saa ohjausta takaisin hahmoon, tekoäly jää jumiin, hahmo onnistuu menemään seinän tai lattian sisään. [26; 27; 28.]

Seinähyppymekaniikalla tarkoitetaan sitä, kun pelaaja hyppää seinää päin ja pelaajan osuessa seinään pelaaja voi painaa uudelleen hyppynäppäintä, jolloin hahmo lähtee ylös ja vastakkaiseen suuntaan seinästä. Kun on kaksi seinää vastakkain pelaaja voi tehdä sarjassa seinähyppyjä, jolloin pelaaja "kiipeää" ylös korkeammalle. Seinähyppymekaniikkoja on monissa tasoloikkapeleissä, 3D-

ja 2D-peleissä, esimerkiksi Nintendon Mario-peleissä on käytetty tätä mekaniikkaa. [29.] Kuvassa 10 on seinähyppy käytössä.



Kuva 10. Kuvankaappaus Ratchet and Clank 2:n seinähyppytilanteesta.

3 Unity-pelimoottori

Pelimoottori, jota käytettiin tässä projektissa, on Unity 3D-versio 2022.1.23f1, se päivitettiin 2021.1.10f1 versiosta, koska uudemmissa versioissa on lisätty spline-ominaisuus. Unity-pelimoottorissa käytetään C#-ohjelmointikieltä. Unityllä on useampi lisenssi, peruslisenssin on ilmainen, mutta sillä on rajoitus, kun sillä tehty sovellus ylittää tuotoissaan tietyn rajan, täytyy hankkia maksullinen lisenssi. Nämä lisenssit ovat Plus ja Pro. On olemassa myös opiskelijoille suunnattu lisenssi. [30.]

3.1 CharacterController-komponentti

Unityssä on useampi tapa tehdä pelihahmon liikuttaminen, mutta yleisimmät ovat Rigidbody ja CharacterController. Rigidbodyssa on eniten työtä mekaniikkojen toteutuksessa, mutta on myös joustavampi kuin CharacterController. Syy miksi tässä projektissa käytetään Rigidbodya, on CharacterControllerin osumialuetta ei voi kääntää kuin Y-akselilla, mutta projektissa pitää saada käännettyä sitä sekä Z- ja X-akseleilla useammasta syystä. Yksi näistä mekaniikoista inspiraationa otettu mekaniikka on Ratchet and Clank-sarjasta magneettikengät, jotka halutaan tehdä tähän peliin, jotka mahdollistaisivat tietyillä seinillä tai katossa kävelyn. [31.]

3.2 Rigidbody-komponentti

Rigidbody-komponentti pistää esineen, jolla on tämä Unityn fysiikkamoottorin alaiseksi, tämä tarkoittaa sitä, että Rigidbody-komponentin omaavat esineet voivat työntää toisiaan. Tähän komponenttiin vaikuttaa esineen koko, isolla esineellä olisi siis suurempi ilmanvastus, kun se putoaa, mikä tarkoittaa sitä, että se putoaa hitaasti verrattuna esineeseen, joka on pieni. Unity määrittelee esineen koon esineen mittakaava arvosta. Rigidbodyssa on oma painovoimamekaniikka mutta tämän painovoiman suunta on sama kaikilla Rigidbody-objekteissa. Rigidbodyn painovoima suuntaa voidaan muuttaa, mutta tämä vaikuttaa kaikkiin Rigidbody-objekteihin. [32.]

3.3 Input Manager

Input Manageri on yksi tapa ottaa Unityssä näppäinten sisääntulon ja käyttää sitä koodissa, vaikka hyppynapin asettamiseen. Input Managerin huono puoli on se, kun peli rakennettu, sen asetuksia ei voi muuttaa enää kuin uudelleen rakentamalla. Tämä on huono, kun halutaan tehdä pelistä esteetön. Input Manageria voidaan käyttää käyttämällä koodissa `Input.GetButton`, kun halutaan saada tosi, kun nappia painetaan. Napit määritellään Unityn Editorissa project settingissä antamalla napille nimi, yksi tai useampi keycode, esimerkiksi eteenpäin on määritelty siten, että sen nimi on `Vertical`, positiiviset napit ovat `W`- ja nuolinäppäin ylös, `S`- ja nuolinäppäin alas on negatiiviset. `Input.GetKey` ei käytä Input Manageria samalla tavalla ja melkein ajaa saman asian, mutta vaatii saada napin nimen suoraan antamalla keycode-version siitä, eikä se pysty ottamaan ohjaimesta sauvaohjaimen asentoa. [33.]

3.4 Input System

Unityn Input System on Input Managerin korvaava sisääntulo, mutta ei ole asennettuna Unityyn oletuksena, vaan pitää asentaa erikseen. Tämä uusi systeemi sallii käyttäjän tehdä näppäinten vaihtoja toisin kuin edellinen Input Manager. Tässä systeemissä on ominaisuus nimeltään Action Maps, tällä pystyy määrittelemään mitä, näppäin tekee tietyssä tilanteessa esimerkiksi valikossa ja ajoneuvolla ajaessa sama nappi tekee eri asian. [34.]

3.5 Spline-komponentti

Splinet ovat käyriä, joissa on solmuja, joilla voidaan määrittää käyrien muodon. Unityssä on Spline-komponentti, ne toimivat versiosta 2022.2 eteenpäin. Tämä mahdollistaa käyrien tekemisen peliin. Tällä Spline-komponentilla pystyy tekemään mutkittelevan tien tai mutkittelevan putken, laittaa peli esineen menemään tai animoida esineen menemään pitkin splineä. [35.]

4 Pelitestauksesta

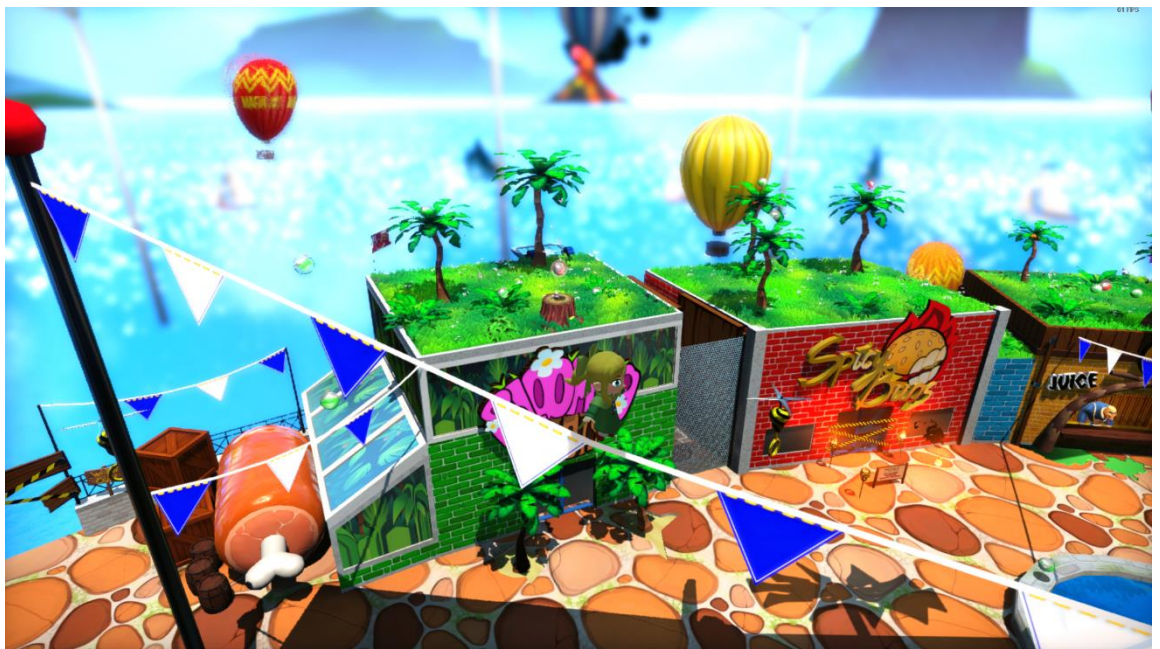
Pelistä voidaan testata useampia asioita, kuten onko käyttöliittymä oikean näköinen, toimiiko peli eri laitteilla ja millaista peliä on pelata [36]. Peliä on hyvä testata jo niin pian kuin sitä voi pelata, pelin ei tarvitse näyttää vielä testausvaiheessa valmiilta, vaan pelaaja hahmon tilalla voi olla vain harmaa kuutio, vaikka valmiissa pelissä siinä pitää olla ihmismäinen hahmo. Jos peli testauksen aloittaa liian myöhässä voi olla, että jotain oleellista vikaa ei ehdi korjata enää.

Pelattavuuden testauksessa voidaan kohdentaa tiettyyn asiaan pelissä, kuten miltä hahmo tuntuu pelata. Tällä tavalla saa tietää, että onko tämä tietty ominaisuus hyvä. Sitten on sokkotestaus, siinä testaajalle ei anneta tietoa, mitä pelissä tulee testata, vaan testaaja pelaa peliä. Näissä pelin kehittäjä voi seurata testaajaa joko vierestä tai haastatella testaajaa pelin lopuksi. Pelin tekijä on huono testaaja, koska tekijä tietää, miten peli toimii ja tietää kaiken pelistä. Tekijän on hyvä olla puuttumatta peliin, kun testaaja pelaa peliä, vaikka tekisi mieli. [37.]

5 Toteutus

Tämä peli projekti alkoi 2019. Ideana oli jo alun perin tehdä peli, joka ottaa inspiraatiota pelattavuudeltaan Sly Cooper -pelisarjasta Sly-hahmoa. Tämä hahmo siis pystyisi muun muassa kiipeilemään putkia pitkin ylös, kävelemään köyttä pitkin ja hyppäämään terävän pisteen päällä. Yksi mekaniikoista on juoksu, joka lisää hahmon nopeutta, mutta mahdolliset viholliset voivat huomata tämän hahmon. Peliin tehdään myös toinen hahmon, joka ottaa inspiraatiota Ratchet and Clank-sarjasta Ratchet-hahmon liikkeistä, mutta pääasiassa hänen ampumismekaniikoistansa. Tämän hahmon liikkeet olisivat railgrind, seinillä kävely, rakettikengät ja heittokoukku. Railgrind-mekaniikassa hahmo on raiteella ja pitää varoa esteitä hyppäämällä yli, toiselle raiteelle tai tuhoamalla se. Toinen mekaniikka on seinillä kävely, tämän avulla hahmo pystyy kävelemään tiettyjä pintoja pitkin, joita on seinillä ja katoilla. Kolmas mekaniikka on rakettikengät, nämä kengät mahdollistaisivat nopean liikkumisen pitkin kenttiä. Heittokoukkumekaniikalla pystyy ylittämään esteitä heittamalla tai vetämällä itsensä esteelle tai esteen itseensä luokse.

Pelin pääasialliset inspiraation lähteet ovat PlayStation 2 -pelit, mutta näitä samanlaisia mekaniikkoja on myös muissa peleissä tai pelisarjoissa. Näitä saman tapaisia mekaniikkoja on Keen Softwaren Space Engineer -pelissä oleva magneettikenkämekaniikka [38], Gears for Breakfastin Hat In Time -pelin köydellä kävely, kuten kuvassa 11 on pelin päähahmo kävelemässä köydellä.



Kuva 11. Kuvankaappaus Hat In Time köydellä kävely -mekaniikka.

Pelaaja hahmon liikuttaminen tapahtuu RigidBody-peliobjektin avulla, jota käskytetään koodin avulla sen sijaan, että käytettäisiin CharacterController-komponenttia, koska CharacterController-komponenttia ei voi kääntää kuin Y-akselilla, hahmon pitää pystyä kääntymään kaikilla aksleilla, jotta seinälläkävely toimisi oikein. Pelihahmoa ohjataan joko näppäimistön ja hiiren tai peliohjaimen avulla. Pelaaja voi vaihtaa koska tahansa näppäimistöstä peliohjaimeen tai päinvastoin. Oikealla analogisella sauvalla tai hiirellä ohjataan kameraa.

5.1 Pelaajan hahmon liikkuminen

Hahmoa liikutetaan näppäimistön WASD-näppäimillä tai peliohjaimella vasemmasta sauvaohjaimesta Z- ja X-akseleilla eli eteen, taakse, vasemmalle ja oikealle. Suunta mikä lasketaan eteenpäin suunnaksi, riippuu kameran suunnasta. Kun W-painiketta tai sauvaohjainta työnnetään eteenpäin, hahmo kulkee pois päin kamerasta ja tämä suunta otetaan kameran paikallisesta Y-akselin suunnasta suhteessa pelaajaan.

Hahmot voivat joko juosta tai kävellä, tämä tapahtuu painamalla joko Shift-näppäintä, RB-näppäintä, eli right bumper, Xbox-ohjaimesta tai R2-näppäintä PlayStation-ohjaimesta. Kuvassa 12 olevassa Xbox-ohjaimessa on 7 kohdalla RB-näppäin.



Kuva 12. Kuva Xbox-ohjaimesta. [39]

Kun painetaan näppäintä, joka on määritetty juoksupainikkeeksi, pelaajan hahmon nopeuteen lisätään koodissa kerroin x , tämä tekee hahmon liikkumisesta x kertaa nopeamman. Syy miksi pelaaja ei haluaisi juosta koko ajan on se, että juoksu pitää meteliä ja herättää vihollisten huomion. Tämä tapahtuisi siten, että kun pelaaja juoksee tarpeeksi lähellä vihollisia, vihollinen lähtee menemään kohti pistettä, jossa pelaaja juoksi viimeksi. Samanlaisia mekaniikkoja on Skyrimissa jollain asteella ja myös Hat In Time -peleissä. Kuvassa 13 on tästä pseudokoodi.

```

1
2 FixedUpdate
3 {
4     bool pelaaja maassa tarkistus
5
6     if (juoksu painike painettu)
7     {
8         juoksuKerroin = 2;
9     }
10    else
11    {
12        juoksuKerroin = 1;
13    }
14
15    liikkumis vektori = liikkumissuunnan laskeminen * nopeus * juoksuKerroin
16    painovoima = painovoiman laskeminen
17
18    if(ei ole tarttunut kielekkeelle tai liikkuminen ei ole estetty)
19    {
20        if (Seinähyppy menossa) {liikkumis vektori = seinähypyn suunta}
21
22        liikkumisSuunta = liikkumis vektori + painovoima
23
24        rigidbodyn kiihtyvyys = liikkumisSuunta
25    }
26 }

```

Kuva 13. Pseudokoodia pelaajan liikkumisesta ja kuinka juokseminen vaikuttaa pelaajahahmoon.

Hahmon painovoima toimii hahmon X- ja Z- akselin suunnan perusteella, sen sijaan kuin mikä on maailman alaspäin suunta, eli maailman Y-akselin suunta. Tämä mahdollistaa sen, että kun hahmoa käännetään seinää vasten, seinä on nyt lattia ja edellinen lattia on nyt seinä.

Hahmot pystyvät hyppäämään ja tekemään tuplahypyn ilmassa painamalla välilyöntinäppäintä tai ohjaimesta hyppynäppäintä. Koodissa on määrätty, kuinka monta kertaa hahmo pystyy hyppäämään ilmassa. 0 tarkoittaa, ettei pysty lainkaan hyppäämään ilmassa, kun taas x määrä mahdollistaa x määrän hyppyjä ilmassa. Jos pelaaja painaakin sekunnin osan verran liian myöhään hyppynappia, kun hyppää pudotuksen yli, pelaajahahmo pystyy hyppäämään vielä ilmassa. Tämä ominaisuus on coyote time, nimi tulee Maantiekittäjä-piirretystä, kun kojootti leijuu hetken il-

massa, kunnes tajuaa, että on ilmassa ja alkaa sitten vasta pudota. Kuvassa 14 näytetään pseudokoodin avulla, miten tämä on toteutettu. Tällainen ominaisuus on esimerkiksi Celeste-pelissä, se on pieni ominaisuus, mutta vaikuttaa pelin tuntuun. [40.]

```

1  Public GravityHandler
2  {
3      if (notGrounded && !jumpButtonPressed)
4      {
5          putoaa
6          if(coyoteTimeLeft > 0 )
7          {
8              coyoteTimeLeft -= sekunnit
9          }
10     }
11     else
12     {
13         Ylämäki ja rappusten kävely logiikka
14     }
15
16     If(jumpButtonpressed)
17     {
18         if (grounded){ Hyppää }
19         else if(coyoteTimeLeft > 0 ) { Hyppää}
20         else if (airJumpCount < maxAirJumps ) { Hyppää }
21     }
22
23 }

```

Kuva 14. Pseudokoodi kuvaa, miten hyppy, coyote time ja ilmahyppy toimivat.

5.2 Liikkumistyylin muuttuminen ja taistelussa liikkuminen

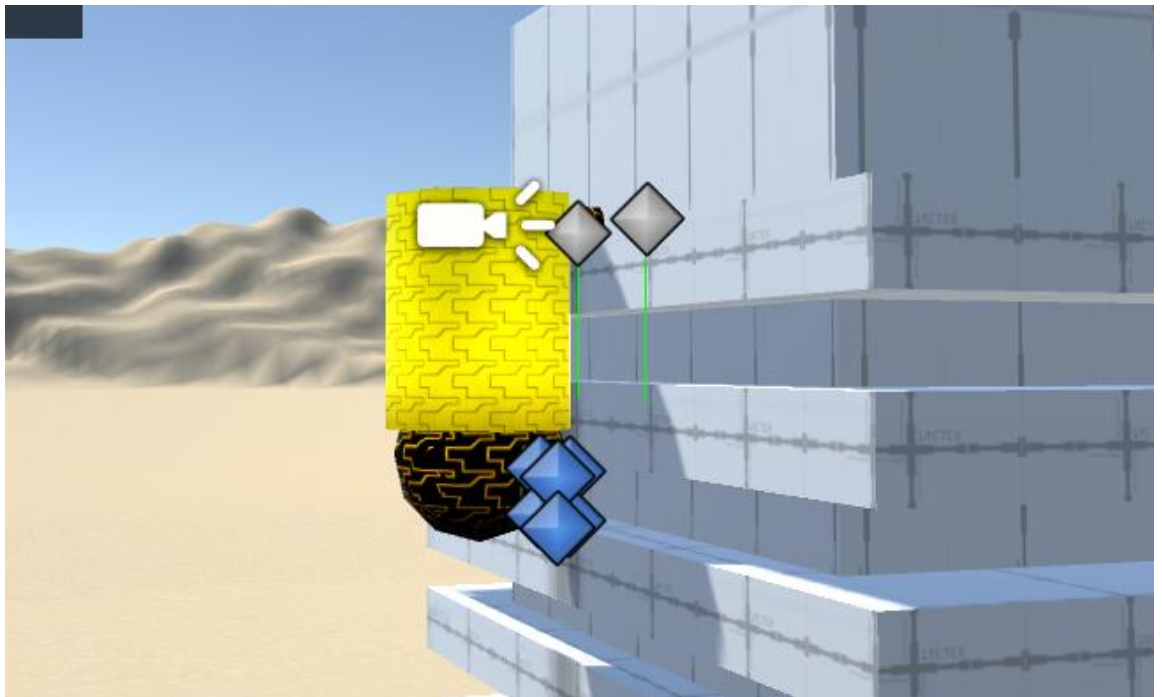
Kameranlukitus-mekaniikan inspiraationa toimi Ratchet and Clank -pelisarjasta. Tämä tapahtuu painamalla kameralukituksen näppäintä pohjassa, jolloin kamera lukittuu pelaajan taakse ja pelaajahahmo katsoo koko ajan kamerasuuntaan, mikä helpottaa ampumista.

Projektissa on useampi pelattava hahmo. Yksi hahmo pystyy käyttämään köysiä ja muita samantapaisia liikkeitä, mutta toiset hahmot eivät voi käyttää näitä. Hahmojen vaihto voidaan tehdä miltei koska tahansa ja voi myös tapahtua automaattisesti tilanteen mukaan. Tämä mekaniikka on verrattavissa kaikkiin kolmeen peliin.

Lyömisen hakeutuminen vihollisiin tapahtuu, kun pelaaja lyö vihollisia. Kun pelaaja painaa lyöntinappia hahmo kääntyy kohti vihollista, jos vihollinen tarpeeksi lähellä ja on pelaaja nähden tarpeeksi edessä.

5.3 Ympäristöstä riippuvaiset mekaniikat

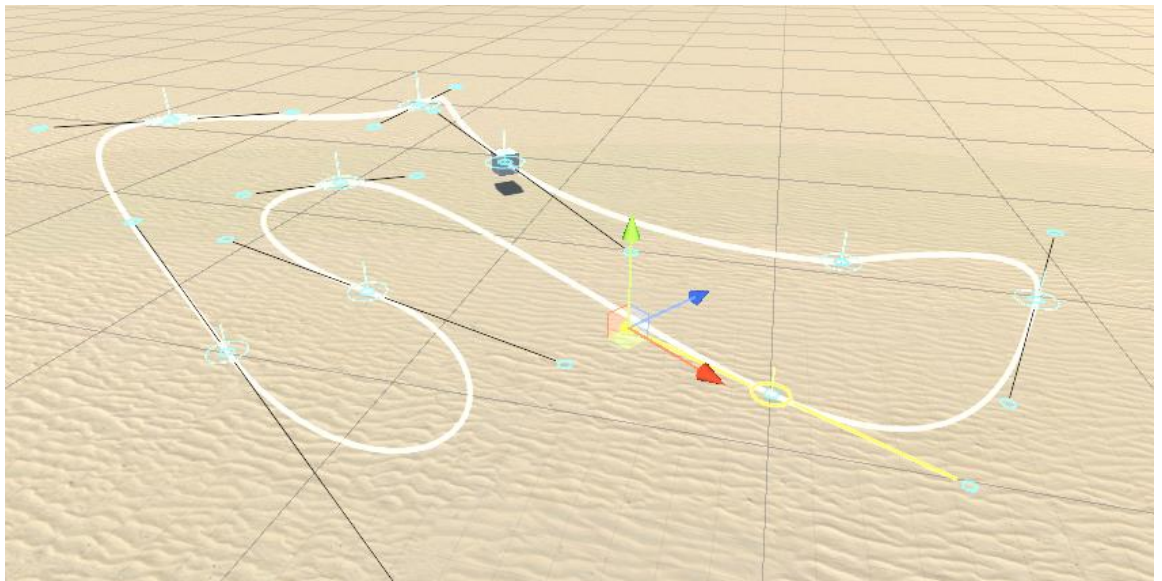
Kaikissa kolmessa inspiraatiopelissä hahmot voivat tarttua kielekkeen reunalle ja hypätä siitä ylös, mikä on toteutettu tässä projektissa siten, että pelaajahahmossa on kaksi tyhjää peliobjektiä. Näistä peliobjekteista pistetään Unityn Raycastin alaspäin tarkastamaan, onko niiden alapuolella kieleke, kuvassa 15 näkyy vihreällä piirretty debug raycasti alaspäin harmaista peliobjekteista. Samoista peliobjekteista on myös Unityn Sphercast tarkistamassa, että nämä pisteet eivät ole seinän sisällä, koska muuten hahmo pystyy tarttumaan seinään.



Kuva 15. Kuva pelaajahahmosta kielekkeelle tarttuneena, jossa näkyy debug-objektit.

Sly-peleissä oleva Sly-hahmon köydellä tai jollain muulla pitkällä ohuella alustalla kävely mekaniikan tekeminen peliin. Tämä "lukitsisi" hahmon köydelle, joten hahmo pystyy liikkumaan vain köyttä pitkin, kunnes pelaaja hyppää, köysi katkeaa tai pelaaja hahmo ottaa osuman jostain. Tämä tehtiin tekemällä solmuja, joiden välillä liikkuu piste. Kun pelaaja ei ole raiteella, piste etsii solmun, joka on pelaajaa kohden lähin, tämän jälkeen etsitään toiseksi lähin solmu, sitten lasketaan lähin kohta näiden kahden solmun välillä, jonne sitten piste siirretään. Kun pelaaja on aktivoinut

tämän pisteen, hahmo siirtyy köydelle. Pelaaja pystyy sitten liikuttamaan hahmoaan pisteen avulla kohti jompaakumpaa solmua ja kun piste saavuttaa solmun, piste ottaa seuraavan solmun. Kun piste on viimeisen solmun kohdalla, hahmo pysähtyy, tässä tilanteessa pelaajahahmolla voi joko hypätä pois köydeltä tai lähteä toiseen suuntaan. Tämä toteutus korvattiin Unityn splineillä, mutta toiminnaltaan se on sama. Syy miksi splinejä ei käytetty heti alussa oli, että Unityssä ei ollut splinejä, vaan ne lisättiin myöhemmin 2022.2-versiossa. Samantapainen köydellä kävely -mekaniikka on myös Hat In Time -pelissä, jossa tarvitsee vain hypätä köyden päälle. Sly-peleissä Sly-pelihahmo pystyy kiipeämään putkia ja köysiä pitkin, tässä on sama periaate kuin köydellä kävelyssä, mutta animaatio on erilainen. Yksi mekaniikka, joka on ottanut inspiraatiota ninjatorni-hyppy liikkeestä Sly-peleistä. Tämä liike on saman kaltainen kuin köydellä kävely, mutta sen sijaan pelaajahahmo pystyy hyppäämään terävän pisteen päälle ja pystyy seisomaan siinä ja kääntymään 360 astetta, samat säännöt kuin köydeltä poistumiseen eli hyppy tai pelaajahahmo ottaa osuman. Sama toiminta on, kun pelaaja lukitaan pisteeseen, kun pelaaja on pisteessä, jolloin hahmo ei voi liikuttaa, pelaaja voi yhä lyödä, kääntyä tai hypätä pois tästä pisteestä, kun haluaa, jolloin hahmon liikkuminen jatkuu normaalisti. Kuvassa 16 on spline, joka toimii railgrind-mekaniikan pohjana.

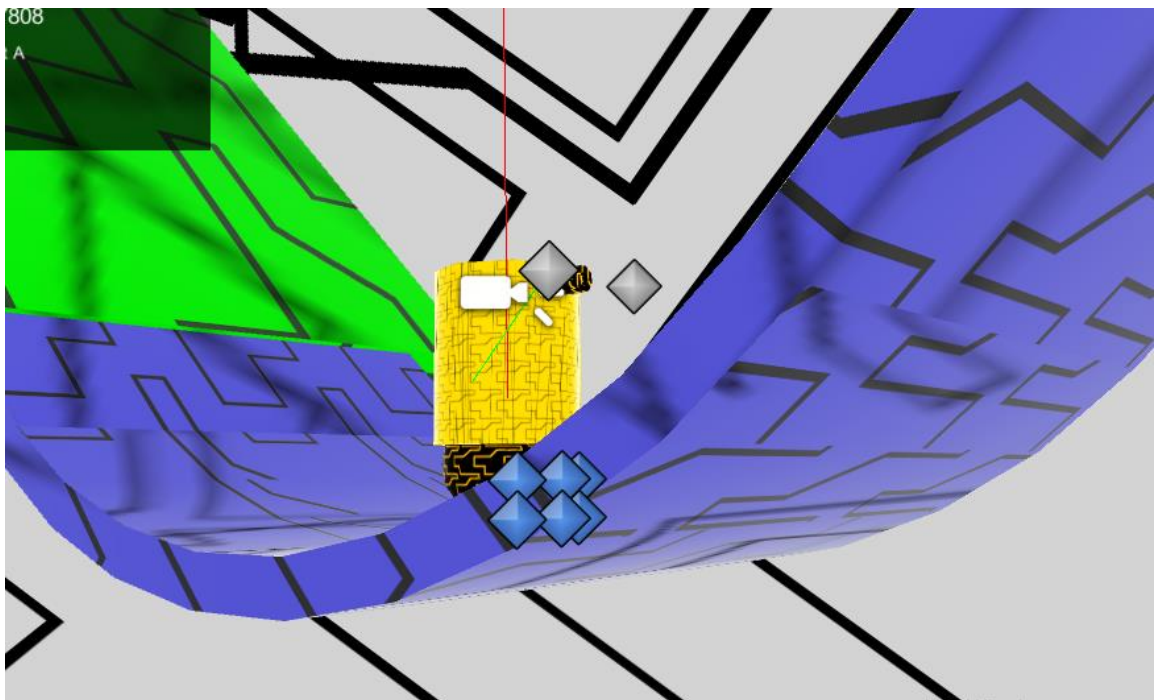


Kuva 16. Railgrind-mekaniikan spline.

Railgrind-pelimekaniikka on kuin köydellä kävely, mutta sen sijaan, että pelaaja määrää kulkuvauhdin, pelaaja kulkee raiteen määräämällä nopeudella eteenpäin. Jos seuraava solmu alempana kuin edellinen, hahmo kulkee nopeammin raiteella tai jos seuraava on korkeammalla, pe-

laaja hahmon vauhti hidastuu raiteella. Railgrind mekaniikkana on melko yleinen, ja tämä mekaniikka löytyy, Ratchet and Clank-, Jak and Dexter- ja 3d Sonic-peleistä. Kun kaksi raidetta kulkee rinnakkain, pelaaja voi hypätä toiselle raiteelle tarvittaessa.

Magneetikengät ovat käytössä tietyissä tilanteissa Rathcet and Clank -pelisarjassa, ne otetaan käyttöön automaattisesti, kun pelaajaa astuu tiettyjen pintojen päälle. Samanlainen mekaniikka on tässä projektissa, kun pelaaja astuu tietyn pinnan päälle, hahmo kääntyy tämän pinnan mukaan ja on aina 90 asteen kulmassa tästä pinnasta. Myös pelaajan painovoiman suunta muuttuu samalla kun pelaaja kääntyy pinnan mukaan, eli kun pelaaja hyppää, pelaaja putoaa kohti pintaa, jonka päällä oli. Kun pelaaja menee liian kauas pinnasta tai astuu pois pinnan päältä, pelaajahahmon painovoima muuttuu normaaliksi, eli y-koordinaatti on ylös ja alas jälleen. Kuvassa 17 pelaajahahmo on testikentässä pinnalla, joka kääntää pelaajaa. Saman tapainen mekaniikka on Mario Galaxyssa ja Sonic Adventuressa.



Kuva 17. Pelaaja on pinnalla, joka aktivoi painovoiman suunnanvaihdon.

Kun pelaajaa aktivoi rakettikengät, pelaajahahmo liikkuu koko ajan eteenpäin. Hahmo liikkuu nopeammin kuin juoksu ja pystyy käyttämään rampeja. Tämä mekaniikka muuttaa pelaajahahmon liikkumisen lähemmäs raketia tai ajoneuvoa kuin kävelyä. Saman tapainen mekaniikka on Jak and Dexter -pelissä leijulauta, Hat In Timessa skootterimerkki. Kun pelaaja käyttää tätä, hahmo ei pysty lyömään tai ampumaan, keskeyttämättä rakettenkiä. Kun pelaajaan osuu seinään tai ottaa

vahinkoa, rakettikengät poistuvat käytöstä, kunnes pelaaja ottaa ne käyttöön uudestaan. Käytännössä hahmo liikkuu koko ajan eteenpäin ja painovoiman arvoa pienennetään rakettikengien käytön ajaksi.

Swingshot-mekaniikka on Ratchet and Clank -pelissä käytetty mekaniikka, mutta se tunnetaan paremmin heittokoukkuna eli grappling hook. Pelissä on tiettyjä pisteitä, jotka ovat ilmassa, joihin pelaaja pystyy ampumaan pisteeseen vaijerin, jota käyttäen pystyy heiluttaa itsensä, vaikka rotkon toiselle puolelle, vetämään itsensä toiselle puolelle tai vetämään jotain puoleensa. Muita pelejä, joissa saman tapaisia grabbling hook -mekaniikkoja on Hat In Time tai Sonic-fanipelissä nimeltä Sonic Omens. Pelissä on peliobjekteja, joita lähestyessä pelaaja voi sitten käyttää painamalla näppäintä. Pelaajahahmo liikkuu suoraan kohti tätä peliobjektia ottamalla vektorin peliobjektin ja pelaajahahmon välillä, kunnes on tarpeeksi lähellä. Tämän aikana liikkuminen on estetty tavallisessa liikkumiskoodissa.

5.4 Projektin jatkokehitys

Pelissä on toimivia mekaniikkoja, mutta ne ovat vielä kesken, joitakin mekaniikkoja ei ole edes alettu tekemään ja on vain suunnitteluasteella, kuten uiminen. Projektista on tarkoitus tehdä myöhemmin kokonainen peli, jossa on tarina, kunnolliset 3D-mallit, tekstuurit, vihollisten tekoäly ja loput mekaniikoista, jotka eivät liity liikkumiseen. Peliin ei tule ääninäyttelyä. Peli tullaan julkaisemaan osissa, jos aikaa riittää projektin jatkamiseen. Projekti saattaa myös jäädä tähän tai projekti tehdään uudelleen joko Unityllä tai eri pelimoottorilla.

Projektia tullaan testaamaan pelin kehittyessä väliajoin, että pystytään huomaamaan uusista tai vanhoista mekaniikoista, ne eivät toimi oikein tai eivät sovi peliin. Testauksessa saattaa tulla myös vastaa vika, josta pelaajat saattavat pitää ja sen vian korjaaminen voi olla huono asia pelaajien mielestä. Tästä viasta voi tulisi peliin pysyvä mekaniikka. Projektiin tehtiin yksi testaus, mutta tämä kysely epäonnistui pelissä olleen vian vuoksi. Pelin loputtua, kun pelaaja poistuu pelistä, pelin piti avata kyselylomake selaimeen. Tämä ei toiminut, koska peli jumittui päävalikkoon pala- tessa, mikä aiheutti sen, ettei pelaaja voinut sammuttaa peliä oikein, vaan piti pakottaa sammutus. Osa testaajista halusi antaa palautteen suullisena tai tekstinä sen sijaan.

Peliä tullaan testaamaan samalla tavalla: pelistä laitetaan versio nettiin ja pyydetään ihmisiä ko- keilemaan peliä. Tämän jälkeen testaajat täyttävät netissä kyselylomakkeen, jossa kysellään, oliko

tietty mekaniikka hyvä, myös toisia asioita kysellään. Myös testaajan pelaamista tullaan katso-
maan, joko katsomalla vierestä tai etänä.

6 Pohdinta

Erikoisliikkeet peleissä vaativat sen, että pelissä toimiva perusliikkuminen, koska nämä, nämä johdetaan niistä tai että pelaaja pääsee edes käyttämään näitä liikkumismekaniikkoja. Ilman kunnollista pohjaa erikoisliikkumisesta ei ole hyötyä. Monet eri liikkumismekaniikat ovat riippuvaisia ympäristöstä, kuten onko ympäristössä jotain, mikä toimii kuin tikkaat, joita pitkin voisi kiivetä tai raidetta, jota pitkin railgrindata. Nämä ympäristöstä riippuvaiset mekaniikkojen esineet voivat näyttää melkein miltä tahansa, esimerkiksi tikkaina voi toimia rännin putki tai köysi, jotka roikkuvat katon reunasta tai raiteena voi olla junan raiteet, portaiden kaide tai puunrunko.

Peliprojektia tehdessä on hyvä pitää myös silmällä, onko tämä mekaniikka jo olemassa valmiiksi tehtynä tai tullut projektin aikana päivityksen mukana pelimoottoriin, varsinkin kun projektin tekeminen menee useammalle vuodelle. Tämä huomattiin vasta tätä opinnäytetyötä tehdessä. Tämä kyseinen ominaisuus oli Input System, jonka olemassa oleminen huomattiin, kun Input Managerista etsittiin tietoa. Tähän projektiin oli myös tarkoitus tehdä muitakin mekaniikkoja, joita ei ole tässä mainittu, mutta niitä ei edes aloitettu. Yksi näistä mekaniikoista oli uiminen ja sukeltaminen.

7 Yhteenveto

Tässä aiheena oli käydä läpi, kuinka peliprojektin hahmon liikkuminen tehtiin keskittyen erikoisliikkumiseen, kuten magneettikenkä-mekaniikkaan. Nämä asiat tehtiin enemmän ohjelmoinnin näkökulmasta kuin pelisuunnitelmallisesti. Mekaniikat toteutettiin pikemmin silmämääräisesti ja tuntuman perusteella kuin tutkimalla inspiraation lähteiden koodia.

Pelaajahahmon liikkuminen vaihtelee pelien kesken, peligenre antaa suuntaa antavaa tietoa, miten hahmo voi liikkua pelissä, mutta ei varmaa tietoa. Tasohyppelypelien määritelmään kuuluu, miten hahmo liikkuu. Näissä peleissä esteiden ylittäminen tai väistäminen osa peliä ja haasteen kuuluu olla siinä, että pääsee pelissä tavoitepisteeseen eikä hahmon ohjaamisessa.

Projekti otti inspiraatioita vahvasti Sony PlayStation 2 -pelisarjoista. Nämä pelit olivat Sucker Punch Productionin Sly Cooper, Insomniac Gamesin Ratchet and Clank, ja Naughty Dogin Jak and Daxter. Näiden pelien mekaniikkoja on myös muissakin peleissä, kuten Segan Sonic- ja Nintendon Mario-peleissä.

Projektissa tuli vastaan asioita, jotka hankaloittivat etenemistä, kuten CharacterController-komponentti kääntyy vain yhdellä akselilla, kun sen pitää kääntyä kaikilla kolmella, X-, Z- ja Y-akselilla. Pelaaja pystyy liikkuttamaan hahmoaan joko näppäimistöllä ja hiirellä tai Xbox-ohjaimella. Taistelussa pelaajalla on käytössä muutama mekaniikka, joka auttaa taistelussa. Jotkin erikoisliikkumismekaniikoista vaatii, että ympäristössä on tietynlainen esine, että sitä liikkumista voidaan käyttää esimerkiksi raide, että hahmo voi railgrindata.

Projektiin tehtiin pelitestausta, mutta tämän testauksen tulosten saanti epäonnistui ja tulokset ovat jääneet analysoimatta. Tämä johtui pelissä olleesta viasta, joka esti kyselylomakkeen automaattisen avautumisen. Testaus hoidettiin antamalla testaajille pelin testattavaksi, jonka jälkeen testaajat vastasivat kyselyyn. Joitakin tuloksia saatiin, mutta nämä tulokset tulivat sanallisina.

Lähteet

- 1 Kempainen Jaakko. Genremetsä – peligenrejen käyttö digitaalisissa palveluissa [Internet] 2012. [viitattu 10.5.2022], saatavilla: <https://www.pelitutkimus.fi/vuosi-kirja2012/ptvk2012-06.pdf>
- 2 Mobygames, (2022) Genre Definitions, [Internet], [viitattu 29.3.2022], Saatavilla: <https://www.mobygames.com/glossary/genres>
- 3 Make use of, (2020) First-Person Games vs. Third-Person Games: What Are the Differences?, [Internet], [viitattu 17.3.2023], Saatavilla: <https://www.makeuseof.com/first-person-games-vs-third-person-games-differences/>
- 4 Jensen K. Thor, Knee Deep in the Dead: The History of First-Person Shooters. [internet] 2022, [viitattu 17.3.2023], saatavilla: <https://www.pcmag.com/news/the-complete-history-of-first-person-shooters>
- 5 McComas Shane: (Arcade) Battlezone [video], [Internet], 2008 [viitattu 24.3.2023], saatavilla: https://www.youtube.com/watch?v=Ctr54kopo8I&ab_channel=ShaneMcComas
- 6 Giant Bomb, Rocket jump, [Internet], [viitattu 17.3.2023], Saatavilla: <https://www.giant-bomb.com/rocket-jump/3015-20/>
- 7 Id Software Quake, Steam [kuva], [Internet], [viitattu 2.4.2023], Saatavilla: <https://store.steampowered.com/app/2310/Quake/>
- 8 Capcom Co., Ltd., Resident Evil [kuva], [Internet], [viitattu 2.4.2023]. saatavilla: https://store.steampowered.com/app/304240/Resident_Evil/
- 9 Insomniac Games, Ratchet & Clank 2: Going Commando, [käyttöohjeet, s.32] 2004
- 10 Rouse Margaret, Role-Playing Game, [internet] 2020 [viitattu 24.3.2023] Saatavilla: <https://www.techopedia.com/definition/27052/role-playing-game-rpg>
- 11 Divinity: Original Sin 2 [internet], [viitattu 30.3.2023], saatavilla: <https://divinity.game/>

- 12 Play Made Simple, Skyrim Useless Tip # 1 - Jumping in a Forge Levels Destruction [Video] 2011, [Internet], [viitattu 24.3.2023], saatavilla: https://www.youtube.com/watch?v=yyLDtVf4K4s&ab_channel=PlayMadeSimple
- 13 The Elder Scrolls wiki, Whirlwind Sprint, [internet], [viitattu 3.4.2023], Saatavilla: https://elderscrolls.fandom.com/wiki/Whirlwind_Sprint
- 14 Lucroprg, Tactical Retreat - Divinity 2 [video], [Internet], 2017, [viitattu 24.3.2023], saatavilla: https://www.youtube.com/watch?v=Zu0A37w4RVU&ab_channel=Lucroprg
- 15 Checkers Prefect, How to Find Easy Teleportation in Divinity: Original Sin 2, [Video] 2017, [Internet], [Viitattu 24.3.2023], saatavilla: <https://youtu.be/nbxkeuq8scs?t=210>
- 16 Career gamers, Your Guide To 3d Platformer Game Design (Clear explanations), [Internet], [viitattu 24.3.2023], saatavilla: <https://careergamers.com/your-guide-to-3d-platformer-game-design/>
- 17 Moby Games, Sly Cooper series, [Internet], [viitattu 29.12.2022], saatavilla: <https://www.mobygames.com/game-group/sly-cooper-series>
- 18 Sucker Punch Productions LLC, Sly 3: Honor among thieves, [käyttöohjeet] 2005
- 19 Insomniac Games, Ratchet & Clank 2: Going Commando, [käyttöohjeet, s.32] 2004
- 20 IGN Wiki guide, Ratchet and Clank: Going Commando Wiki Guide, [internet], [viitattu 3.4.2023], Saatavilla: https://www.ign.com/wikis/ratchet-clank-going-commando/Platinum_Bolts
- 21 Naughty Dog LLC, Jak 3, [käyttöohjeet, s. 28] 2004
- 22 Tv tropes, Springs, Springs Everywhere, [internet], [viitattu 23.3.2023], saatavilla: <https://tvtropes.org/pmwiki/pmwiki.php/Main/SpringsSpringsEverywhere>
- 23 Tv Tropes, Floating Platform, [internet], [viitattu 23.3.2023], saatavilla: <https://tvtropes.org/pmwiki/pmwiki.php/Main/FloatingPlatforms>
- 24 Tv Tropes, Temporary Platforms, [internet] [viitattu 23.3.2023], saatavilla: <https://tvtropes.org/pmwiki/pmwiki.php/Main/TemporaryPlatform>

- 25 Tv Tropes, Bottomless Pit, [internet], [viitattu 23.3.2023], saatavilla: <https://tvtropes.org/pmwiki/pmwiki.php/Main/BottomlessPits>
- 26 Ask a Game Dev – Harder Than You Think: Ladders, [internet], [viitattu 30.3.2023], saatavilla: <https://askagamedev.tumblr.com/post/653352129102315520/harder-than-you-think-ladders>
- 27 Baker Iain 'Sessile_Nomad', Ladders in First-Person Video Games - Rant and Analysis, 2022 [internet], [viitattu 30.3.2023], saatavilla: <https://www.nomadsreviews.co.uk/post/ladders-in-first-person-video-games-rant-and-analysis>
- 28 The Exciting History of Ladders in Video Games, 2020 [internet], [viitattu 30.3.2023], saatavilla: <https://www.safetyfabrications.co.uk/news/exciting-history-ladders-video-games>
- 29 Tv tropes, Springs, Wall Jump, [Internet], [viitattu 30.3.2032], saatavilla: <https://tvtropes.org/pmwiki/pmwiki.php/Main/WallJump>
- 30 Unity. (2021) Compare Unity Plans: Pro vs Plus vs Free. [internet], [viitattu 29.11.2021], saatavilla: <https://store.unity.com/compare-plans>
- 31 Unity Technologies, (2021) Unity Documentation, [internet] [viitattu 30.11.2021], saatavilla: <https://docs.unity3d.com/ScriptReference/CharacterController.html>
- 32 Unity Technologies, (2021) Unity Documentation, [internet], [viitattu 30.11.2021], saatavilla: <https://docs.unity3d.com/ScriptReference/Rigidbody.html>
- 33 Unity Technologies, (2021) Unity Documentation, [internet] [viitattu 30.11.2021], saatavilla: <https://docs.unity3d.com/Manual/class-InputManager.html>
- 34 Unity Technologies, (2021) Unity Documentation, [internet] [viitattu 30.11.2021], saatavilla: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.5/manual/index.html>
- 35 Unity Technologies, Building better paths while maintaining creative flow with Splines in 2022.2, [internet], [viitattu 30.3.2023], saatavilla: <https://blog.unity.com/technology/building-better-paths-with-splines-in-2022-2>

- 36 QAbLe, 7 Different Types of Game Testing Techniques [internet], [viitattu 5.4.2023], saatavilla: <https://www.qable.io/7-different-types-of-game-testing-techniques/>
- 37 Brackeys, HOW TO PLAYTEST!, [video] [internet], [viitattu 5.4.2023], saatavilla: https://www.youtube.com/watch?v=b9U1dcUS_-w&ab_channel=Brackeys
- 38 Space Engineers Wiki, Magnetic Boots,[internet] [viitattu 30.12.2022], saatavilla: https://spaceengineers.fandom.com/wiki/Magnetic_Boots
- 39 Microsoft, Get to know the new Xbox Wireless Controller [Kuva], [Internet], [viitattu 2.4.2023], saatavilla: <https://support.xbox.com/en-US/help/hardware-network/controller/get-to-know-your-xbox-series-x-s-controller>
- 40 Martin Wood, How Celeste's "Coyote Time" Mechanic Elevates the Platforming Experience, 2022 [internet], [viitattu 30.3.2023], saatavilla: <https://gamerant.com/celeste-coyote-time-mechanic-platforming-impact-hidden-mechanics/>