

Pelituotteen kehityksen elinkaari

Otto Uusihakala



Tekijä(t) Otto Uusihakala	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Pelituotteen kehityksen elinkaari	Sivu- ja liitesivumäärä 42+0
Opinnäytetyön otsikko englanniksi Game Product Development Life Cycle	
<p>Opinnäytetyön tavoite on kertoa ohjelmistokehityksestä kokeneelle mutta pelien kehittämisestä kokemattomalle miten pelituotteen kehitys etenee alusta loppuun. Opinnäyte selvittää mitä toimitettavia tulisi saada aikaiseksi missäkin vaiheessa kehitysprojektin elinkaarta.</p> <p>Opinnäytetyö käy läpi tarvittavat elinkaaren vaiheet esituotannosta jälkituotantoon, tarvittavan dokumentaation ja eri versiot pelistä. Työssä käydään myös läpi aikataulun laatimista, kehitystyön ulkoistamista, työroolit, tarvittavat taidot ja tiimin kokoonpanot. Digitaalisten pelien historiaan tehdään lyhyt katsaus ja selitetään miksi pelien kehittäminen on yleistynyt lähiaikoina.</p> <p>Opinnäytetyö ei perehdy tekniikkaan eikä teknisiin päätöksiin, muuta kuin antaa neuvoja siitä, mitä pitäisi ottaa huomioon teknisiä päätöksiä tehdessä. Työssä ei myöskään perehdytä rahoitukseen, markkinointiin ja levitykseen. Samoista syistä taiteen luontiin ja pelisuunnitteluun ei perehdytä. Opinnäytetyö antaa vain joitakin neuvoja miten pelisuunnittelua kannattaa lähestyä kaupallista peliä kehittäessä.</p>	
Asiasanat Peliala, peliteollisuus, elinkaari, projektinhallinta, ohjelmistokehitys, ohjelmistosuunnittelu	

Author(s) Otto Uusihakala	
Degree programme Bachelor of Business Information Technology	
Report/thesis title Game Product Development Life Cycle	Number of pages and appendix pages 42+0
<p>The aim of the thesis is to educate the reader on the matters of the game development project life-cycle, its phases and its progression from beginning to end. The thesis explains what deliverables should be completed in each phase of development life cycle.</p> <p>The thesis goes through the different phases of the life cycle from preproduction to postproduction, the required documentation and different versions of the game. Other subjects that are explained are scheduling, outsourcing development of product assets, roles in the development team, necessary skills and team compositions. A short glance is taken at the history of digital games and the recent increase in game development's popularity is explained.</p> <p>The thesis does not go into technical details and choices of technology in projects, but highlights some factors that should be taken into account when making those decisions. Financing, marketing and distribution are not covered. For the same reasons art asset creation process and game design are not covered. The thesis only gives some advice on how to approach game and art design when the goal is to develop a commercial game product.</p>	
Keywords Game Industry, Life Cycle, Project Management, Software Development, Software Design	

Sisällys

1 Johdanto	1
2 Peliteollisuuden lyhyt historia	3
3 Kehityksen elinkaari	6
3.1 Esituotanto	7
3.1.1 Konseptointi	7
3.1.2 Pelin vaatimukset	16
3.1.3 Pelisuunnitelma	20
3.2 Tuotanto	26
3.3 Testaus	28
3.4 Jälkituotanto tai projektin päättäminen	30
4 Tiimi	31
4.1 Työroolit	31
4.1.1 Insinööri	31
4.1.2 Taiteilija	32
4.1.3 Suunnittelija	33
4.1.4 Testaaja	34
4.1.5 Tuottaja	35
4.1.6 Äänituotanto	35
4.2 Tiimikokoonpanot	35
5 Ketterät kehitysmenetelmät	36
6 Rogue Runner -projektin elinkaari	38
6.1 Esituotanto	38
6.2 Tuotanto	38
6.3 Jatkokehityssuunnitelmat	39
6.4 Mikä meni hyvin? Mikä meni huonosti?	39
7 Johtopäätökset ja pohdinta	41
Lähteet	43

1 Johdanto

Sitä mukaa kun peleistä on tullut valtavirran suosiossa olevaa viihdettä, pelien kehittäminen on yleistynyt ohjelmistokehityksen muotona. Entistä enemmän kehittäjiä pyrkii jossain vaiheessa kehittämään kaupallisen pelin, olisi se sitten mobiililaitteille tai perinteisemmille pelilaitteille.

2015 kevään Softala III –kurssin Rogue Runner –projektissa tiimimme huomasi hyvin aikaisessa vaiheessa, että ymmärryksemme pelien kehittämisestä oli puutteellista. Teknisen osaamisen puutteen pystyi korjaamaan koulutustilaisuuksilla ja parikoodaamisella, mutta ongelma oli perustavanlaatuisempi, sillä emme osanneet muuttaa ideaa toimivaksi peliksi ja puuttui ymmärrys pelituotteen kehityksen elinkaaresta sekä sen vaiheista. Myöhemmin huomattiin aiheen olevan hyvin suosittu opiskelijoiden keskuudessa ja opinnäytetyön aihe syntyi.

Peliprojektin elinkaari koostuu kolmesta vaiheesta: esituotannosta, tuotannosta ja jälkituotannosta. Esituotannossa ja tuotannossa luodaan itse peli iteratiivisesti kehityssyklillä, mikä koostuu suunnittelusta, toteutuksesta ja testauksesta.

Esituotannossa luodaan konsepti, määritellään vaatimukset ja tehdään pelisuunnitelma. Lisäksi kehitetään prototyyppejä testaamaan pelisuunnitelman kuvaamia ominaisuuksia. Elinkaaren päämäärä on muuttaa abstrakti idea pelistä suunnitelmaksi, millä voidaan toteuttaa peli käytännössä. Suurin osa virheistä tapahtuu esituotannossa, mitkä usein johtavat pelin myöhästymiseen tai sen laadun puutteeseen.

Tuotannossa pelin suunniteltu sisältö ja toiminnallisuus toteutetaan pelituotteeksi. Koska tuotannossa vietetään paljon aikaa, tuotannossa on useampi virstanpylväs, mikä helpottaa hahmottamaan projektin edistymisen. Kun Beeta -virstanpylväs saavutetaan, peli on sisällöltään valmis ja projektissa on vain testausta jäljellä julkaisuun asti. Päämääränä on saada kaikki ongelmat korjattua tuotteessa, jotta se olisi riittävän laadukas julkaisuhetkellä.

Jälkituotannossa arvioidaan projektin menestys ja arkistoidaan projektin tuotokset myöhempää käyttöä varten. Vaikka tiimillä on tähän mennessä jo hyvä käsitys siitä, mikä meni hyvin ja mikä huonosti, on hyvä arvioida projektin menestystä yleisesti ottaen. Jos myöhemmin tarvitsee tarkastaa, miten ongelmat ratkottiin, voi arkistosta kaivaa projektin resurssit esiin, mikä tekee arkistoinnista tärkeää.

Kehitystiimi koostuu kaikenlaisista osaajista, jotka tulevat eri aloilta. On teknisesti lahjakkaat insinöörit, taiteellisesti lahjakkaat taiteilijat, suunnittelijat, jotka ovat mitä vain ja projektihallinnollisesti lahjakkaat tuottajat. Heidän tulee työskennellä hyvin yhdessä saadakseen projektin valmiiksi, ja että lopullinen tuote olisi laadukas.

Ketterät kehitysmenetelmät ovat osoittaneet toimivuutensa perinteisillä ohjelmistokehityksen aloilla, mutta kehittäjillä voi olla hankaluuksia soveltaa niiden oppeja pelien kehittämiseen, missä tarvitaan muutakin kuin ohjelmointia.

Pelien kehittämisessä on selkeitä eroja perinteiseen ohjelmistokehitykseen, mutta samaan aikaan paljon samankaltaisuuksia. Moni voi kuitenkin lähteä pelien kehittämisen pariin väärillä oletuksilla esimerkiksi taiteen ja pelisuunnittelun osalta. Nämä kuitenkin pitää ottaa huomioon ja niihin pitää suhtautua vakavasti, koska ne ovat olennaisia projektin elinkaareissa.

Opinnäytetyössä käytetään muun muassa seuraavia käsitteitä:

Bugi – ohjelmointivirhe mikä löytyy ohjelmistokoodista. Johtaa ohjelmistotuotteen määrittelynvastaiseen toimintaan. Kutsutaan myös ohjelmistodefektiksi.

Pelimuodot – pelin säännöt, mitkä määrittävät miten pelaaja pystyy vuorovaikuttamaan pelin kanssa. Luo pelattavuuden.

Pelattavuus – tapa millä pelaaja on vuorovaikutuksessa pelin kanssa, toimintamalli mikä perustuu pelin sääntöihin. Usein koostuu haasteista mitä pelaaja joutuu ratkaisemaan pelin tarjoamilla välineillä.

Toimitettava tuotos – aikaansaattava projektin tuote, mikä ei kuitenkaan ole itse pelituote. Esimerkkejä toimitettavista tuotoksista on dokumentaatio ja ohjelmistokoodi.

2 Peliteollisuuden lyhyt historia

Pelit ovat pienessä ajassa nousseet valtavirran suosioon yleiseksi ajanvietteeksi. On vaikea sanoa milloin ensimmäinen videopeli ilmestyi, mutta esimerkiksi Brookhavenin kansallislaboratoriossa fyysikko William Higinbotham kehitti vuonna 1958 vieraiden viihteeksi Tennis For Two -pelin, joka käytti oskilloskooppia näyttöpäätteenä (Office of Scientific and Technical Information 1981). Elektronisilla näyttöpäätteillä toimivia pelejä on siis ollut 50-luvulta lähtien.

Videopelit nousivat kuitenkin suosioon vasta 1972, kun Atari julkaisi kuuluisan Pong -videopelin (Stevens 2013). Pong ja sen jälkeen Atarin julkaisemat pelit nostivat arcadet Yhdysvalloissa kansan suursuosioon. Pong päätyi jopa koteihin, muttei ollut ensimmäinen kotikonsoli. Ensimmäinen kotikonsoli oli Magnavox -yrityksen vuonna 1972 julkaisema Odyssey -pelikonsoli, joka oli tappiollinen Magnavoxille (Winter 2012). Vain 500 tuhatta yksikköä myytiin.

Atarin menestys jatkui 1978 julkaistulla Space Invaders -pelillä. Moni osti Atarin 2600 -pelikonsolin 80-luvun alkupuolella vain pelatakseen Space Invadersiä kotona. Atarin suosio johti lukuisten pelikonsolien ilmestymiseen markkinoille. Kysyntä ei kuitenkaan vastannut tarjontaa, ja kupla puhkesi. Vuonna 1983 tapahtui amerikkalaisen peliteollisuuden romahdus. Kun amerikkalaisen peliteollisuuden tuotot olivat korkeimmillaan 3,2 miljardia dollaria vuonna 1983, vuonna 1985 tuotot olivat korkeimmillaan vain 100 miljoonaa dollaria (Stevens 2013). Tuotot olivat laskeneet 97 prosenttia kahden vuoden aikana (Stevens 2013). Pelikonsolien toisen sukupolven aika oli ohi.

Vuonna 1985 Nintendo päätti julkaista Yhdysvalloissa Famicom -pelikonsolinsa. Famicom oli ollut suurmenestys Japanissa, jossa se julkaistiin vuonna 1983. Amerikassa se julkaistiin nimellä Nintendo Entertainment System (NES). Vuonna 1987 Nintendo tuotti 1,5 miljardia dollaria tuottoa. Ero vuoden 1985 tuottojen ja Nintendon vuoden 1987 tuottojen välillä oli huimat 97 prosenttia. (Stevens 2013.)

80-luvulla tapahtunui toinenkin muutos. Kun 70-luvulla kuka tahansa harrastelija pystyi tekemään kaupallisen pelin, nyt julkaisijat (enimmäkseen Nintendo) ja jälleenmyyjät hallitsivat levitystä. 70-luvun vapaus oli luultavasti osasy syy 1983 romahdukseen, mutta nyt taistelu hyllytilasta oli kova ja mahdollisuudet menestyä kaupallisesti pelialalla oli itsenäisille tiimeille erittäin pienet. Nintendo oli pelastanut videopelit, muttei ei tuonut takaisin mahdollisuutta menestyä alalla itsenäisenä kehittäjänä.

Vuonna 1988 Sony teki sopimuksen Nintendon kanssa. Sopimuksen mukaan Sony kehittäisi Nintendon tulevalle Super Nintendo Entertainment Systemille (SNES) CD-ROM -tekniikkaan pohjautuvan lisäosan, jonka nimeksi olisi tullut Play Station (huomaa väli) (Stevens 2013). Tämä ei kuitenkaan koskaan toteutunut: Nintendo liittoutui Philips -laitevalmistajan kanssa, joka oli julkaisemassa Philips CD-i -pelikonsolia (Stevens 2013). CD-i oli Sonyn ja Philipsin yhteistyön tulos ja SNES sisälsi myös Sonyn tekniikkaa, joten Nintendo otti suuren riskin (Stevens 2013).

Insinööri Ken Kutaragi ehdotti Sonyn johdolle, että Sony kehittäisi oman pelikonsolinsa. Tuloksena oli CD-ROM levyjä pelimedian säilytykseen hyödyntävä PlayStation (PSX) (Stevens 2013). PlayStation johti 3D -graafiikoiden yleistymiseen, vaikka Nintendon kilpaileva konsoli Nintendo 64 (N64) oli tehokkaampi (Stevens 2013). Syynä oli CD-ROM: vaikka CD-ROM oli hitaampi kuin N64 ROM -kasetti, siinä oli yli 10 kertaa enemmän tallennustilaa (650MB verrattuna 64MB). Tämä mahdollisti ennen näkemättömän laadun pelin sisäisissä videoissa. Introvideoitten laatu nousi uusille tasoille. Myös sisältöä saatiin enemmän, joten peli pystyi olemaan pidempi. Viidennen sukupolven ylivertainen voittaja oli PlayStation.

PlayStationin suosio jatkui kuudennella sukupolvella PlayStation 2 –konsolilla. Sony PlayStation 2 on maailman parhaiten myynyt konsoli, sillä se on Sonyn 2011 myyntilukujen mukaan myynyt yli 150 yksikköä maailmanlaajuisesti. PlayStation 2:lle on kehitetty yli 3874 peliä. Seitsemännen sukupolven PlayStation 3 taas ei saavuttanut edes kilpailijansa, Xbox 360 –pelikonsolin, tason suosiota, vaikka menestyikin Japanissa paremmin kuin Microsoftin Xbox 360. Kahdeksannen sukupolven PlayStation 4 tosin näyttää voittavan oman sukupolvensa myyntiluvuissa. Vuoden 2015 Sonyn virallisesti julkistamien myyntilukujen mukaan PlayStation 4 on ylittänyt 30,2 miljoonan yksikön myyntiluvun.

2000-luvun –puolella välissä tapahtui muutos tavassa levittää ja myydä pelejä. Microsoft ja pian sen jälkeen Sony julkaisivat omat digitaaliset myynti- ja levitysalustat konsoleilleen (Xbox 360 ja PS3). Samaan aikaan PC:lle ilmestyi Valve –pelitalolta digitaalinen myynti- ja levitysalusta nimeltä Steam, joka on tämän hetken käytetyin digitaalinen pelien levitysalusta. Tämä avasi ovet itsenäisille kehittäjille kaupallisten pelien kehittämiseen. Myöhemmin ilmestyi älypuhelimet ja sitä myötä Applen AppStore ja Googlen Play Store mobiiliapplikaatioita varten.

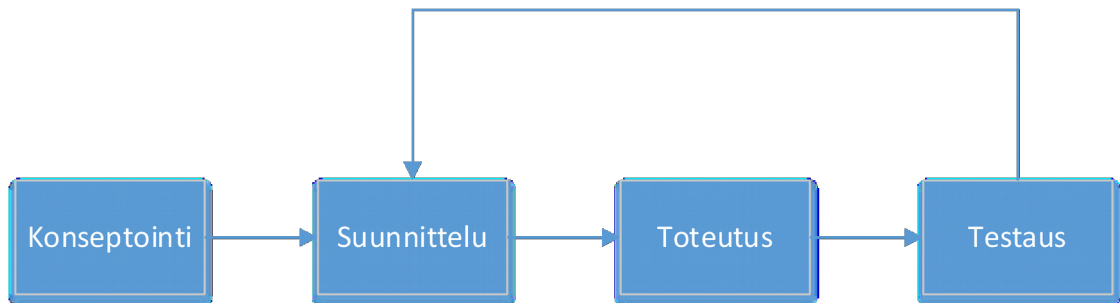
Itsenäisen kehittäjän ei enää tarvinnut kilpailla hyllytilasta. Etenkin PC:llä itsenäinen pelien kehittäminen lähti hurjaan nousuun, koska PC:llä pelien laadunvalvonta ei ollut yhtä tiukkaa kuin konsoleilla ja päivitysten julkaiseminen oli ilmaista, toisin kuin Xbox 360:llä ja

PS3:lla. Jos päivityksen halusi julkaista pelilleen Xbox 360 tai PS3 –konsolilla, joutui maksamaan konsolin valmistajalle 1000 dollarin hinnan. PC:lle ja mobiililaitteille pystyi kehittämään pelejä pienemmällä riskillä ja alhaisemmilla kustannuksilla.

3 Kehityksen elinkaari

Peliprojektin elinkaaren vaiheiden nimet ovat tuttuja elokuvateollisuudesta. Nämä vaiheet ovat esituotanto, tuotanto ja jälkituotanto. Mitä eri vaiheissa tehdään muistuttaa enemmän ohjelmistokehitystä kuin elokuvatuotantoa. Samoin kuin perinteisessä ohjelmistokehityksessä, peliprojekti alkaa määrittelyllä, mistä siirrytään toteutukseen ja sen jälkeen testaukseen.

Jokaisessa vaiheessa, paitsi jälkituotannossa, suoritetaan useampaan otteeseen kehityssykli. Jokaisen ominaisuuden kohdalla tämä kehityksen sykli toistetaan useampaan otteeseen. Konseptointi suoritetaan ainoastaan ensimmäisellä syklillä, kuten kuvasta (kuva 1) näkyy.



Kuva 1. Adams 2010, 45.

Konseptointi suoritetaan peliprojektin alussa. Se on esituotannon ensimmäinen ja erittäin kriittinen vaihe. Sen sisältö määrittelee mitä peli tulee olemaan myöhemmissä vaiheissa. Jos konseptiin ryhdytään tekemään muutoksia myöhemmin esituotannossa tai tuotannossa, paljon aikaisempaa työtä menee hukkaan. Ernest Adamsin (Adams 2010, 45) sanojen mukaan, jos ollaan rakentamassa lentokenttää, siitä ei voi kesken kaiken tehdä hotellia. Jos on päätetty pelin olevan ammuntopeli, se on ammuntopeli loppuun asti. Esituotannon konseptointi on siis vaihe, johon ei palata peliprojektin kuluessa.

1980 -luvulla pelit olivat vielä niin yksinkertaisia, että ne voitiin tehdä loppuun asti laatimalla suunnitelma pelistä ja tekemällä tekninen toteutus heti perään. Nykyään pelit ovat niin monimutkaisia, että peliprojektissa pelin ominaisuuksia pitää kehittää iteratiivisesti jatkuvalla testauksella. Kehityksen aikana palataan sekä suunnitelmaan että toteutukseen useampaan otteeseen muutoksien tekemistä varten. (Adams 2010, 44.)

3.1 Esituotanto

Esituotannossa määritellään mitä pelistä tulee, kuinka kauan sitä kehitetään, kuinka paljon henkilöstöä tarvitaan ja kuinka paljon kehitys tulee maksamaan (Chandler 2014, 5). Toisin sanoen esituotannossa määritellään mitä tehdään, miten ja milloin. Voisi sanoa, että esituotannon aikana tehtävänä on tehdä työn rajaus (scope), joka määrittää millä mittakaavalla peli tehdään. Työn rajauksessa kohtaavat visio ja käytännön toteutus (Portnow 2008).

Esituotanto voi kestää viikosta yli vuoteen. Nyrkkisääntönä on kuitenkin, että esituotanto kestää 10–25% peliprojektin kokonaiskestosta. Jos peliprojekti on mitoitettu kahden vuoden pituiseksi, esituotannon tulisi kestää kahdesta kuuteen kuukautta. Kuuden kuukauden projektissa esituotanto kestää pari viikkoa. (Chandler 2014, 5.)

Peliprojektin työnrajaus ja sisällön suunnitelma, mukaan lukien aikataulu, budjetti- ja henkilöstösuunnitelma, kirjataan usein dokumenttiin, jota kutsutaan pelisuunnitelmaksi (Game Design Document, GDD, tai Game Plan). Tämä dokumentti, joka voi olla kokoelma dokumentteja, jaoteltuna sisällön mukaan, linjaa miten projekti tulee etenemään esituotannon jälkeen. Sen tulisi sisältää pelin konsepti, ominaisuudet, rajoitteet, alustava tekninen- ja suunnitteludokumentaatio, budjettisuunnitelma, henkilöstösuunnitelma ja aikataulu (Chandler 2014, 5).

Tavoitteena on, että esituotannon loppuun mennessä kaikki osakkaat tietävät, mitä pelistä on tulossa ja asiakas on tyytyväinen konseptiin, suunnitelmaan ja prototyyppiin. Kun asiakas on antanut hyväksynnän projektin suunnalle esituotannon lopussa ja hyväksynyt tarvittavan rahoituksen, voidaan siirtyä tuotantoon.

3.1.1 Konseptointi

Esituotanto alkaa konseptoinnilla. Peliä ei voi toteuttaa, ellei tiimillä ole perustavaa käsitystä siitä, millaisia peli tulee olemaan projektin lopussa. Konsepti vastaa tähän kysymykseen: se on visio pelistä, eli sen päämäärä.

Alussa konsepti on hyvin laaja ja epätarkka. Se voi koostua parista lauseesta, eikä ota kantaa asioihin kuten mille alustalle peli tulee olemaan, mitä lajityyppiä se tulee edustamaan tai mitä tehtäviä pelissä tulee olemaan. Se yleensä kuvaa pelaajan toimintaa pelissä.

Konseptia voi lähestyä eräänlaisena ratkaisuna ongelmaan, joka pelaajalla on. Ongelma on kysymys, kuten vaikkapa ”olisiko viihdyttävää ajaa kilpaa konseptiautoilla?” Konsepti on ratkaisu tähän ongelmaan ja vastaus kysymykseen. (Chandler 2014, 131.)

Konseptia voidaan ajatella myös pelaajan unelmana. Pelaajalla on unelma, kuten vaikkapa avaruudessa matkaaminen omalla avaruusaluksella. Kehittäjät toteuttavat unelman peliksi, jolloin peli on ratkaisu unelmaan. Tämä unelma voi liittyä sankaritekoihin, valtaan, voimaan, luomiseen tai johonkin muuhun kokemukseen (Adams 2010, 47). Mitä jos pelaaja haluaa olla valloittaja, kuten Tšingis-kaani? Tässä tapauksessa peli rakentuu valloittamisen ympärille, jossa pelaaja johtaa armeijansa valloittamaan eri maita.

Koko tiimi osallistuu aivoriihiin, joissa pohditaan alustavaa konseptia, mutta lopullisen päätöksen tekee ydintiimi, eli tiimin johto. Tiimin johto koostuu taiteen, suunnittelun ja tekniikan vastaavista sekä tuottajasta, joka on vastuussa projektihallinnosta (Chandler 2014, 132). Pääasiallisesti konseptista on vastuussa suunnittelun vastaava, eli johtava suunnittelija (Portnow 2008). Jos projektille on varattu aikaa kaksi vuotta, alustavan konseptin luomiseen ei tulisi mennä kahta viikkoa enempää (Chandler 2014, 132).

Alustavan konseptin päättämisen jälkeen konseptista hiotaan *lopullinen konsepti*. Se kattaa monta asiaa, kuten

- lajityypin, genren
- kohdeyleisön
- kohdealustan
- tulomallin (revenue model)
- avainominaisuudet
- miljöön
- hahmot
- pelimuodot (game mechanics) ja pelattavuus (gameplay)
- moninpelin, verkko-ominaisuudet

Konseptin osa-alueiden määrittelemiseksi pidetään useampi aivoriihi. Jokaiseen aivoriiheen tulee osallistua ydintiimin lisäksi vain olennaiset tiimin jäsenet (Chandler 2014, 133). Jos aivoriihen aiheena on esimerkiksi visuaalinen tyyli, tulee ottaa mukaan ydintiimin lisäksi taiteilijat ja mahdollisesti pari grafiikkaan erikoistuvaa insinööriä. Aivoriihissä tulee myös olla yhteiset säännöt (Chandler 2014, 133): kritisointi kannattaa kieltää, koska jos kukaan ei uskalla ilmaista mielipidettään kritiikin pelon takia, ei aivoriihestä tule mitään. Jos aivoriihet onnistuvat, ne parantavat tiimin yhteenkuuluvuutta.

Alustavan konseptin jälkeen ensimmäisenä konseptin osa-alueiden listalla on *kohdeyleisö*, joille peli tehdään. He ovat joukko pelaajia, jotka ovat mahdollisesti halukkaita maksamaan pelistä. Kohdeyleisön rooli on suuri etenkin pelaajalähtöisessä suunnittelussa, jossa luodaan hypoteettinen edustaja kohdeyleisölle. Pelaajalähtöisessä suunnittelussa jokainen päätös tehdään tätä hypoteettista edustajaa ajatellen (Adams 2010, 30). Jos ominaisuus ei viihdytä tätä edustajaa, sitä ei toteuteta (Adams 2010, 31). Tämä tekee kohdeyleisöstä hyvin tärkeän osan pelin markkinointia. Tiimiltä tullaan mahdollisesti kysymään rahoituksen hankkimisen yhteydessä, kuka ostaa pelin ja miksi (Adams 2010, 77)? Keitä ostajat ovat ja mitä yhteistä heillä on (Adams 2010, 77)? Nämä ovat hyviä kysymyksiä kohdeyleisön pohdintaan.

Kohdeyleisön määrittämisessä kannattaa varoa binääriajattelua. ”Ainoastaan Ryhmä X pitää asiasta A, eikä kukaan muu” on binääriajattelua (Adams 2010, 73). Kohdeyleisöt usein lomittuvat. Ammuntapeleistä pitävä voi haluta pelata myös uutta roolipeliä, koska siinä on jotain, mikä miellyttää häntä. Kaikkia ei kuitenkaan voi miellyttää. Peli tulee siis suunnitella siten, ettei poissuljeta ketään. On pyrittävä kattavuuteen, ei universaalisuuteen (Adams 2010, 74).

Kohdeyleisön kanssa on myös oltava varovainen. Kattavuus tarkoittaa myös sitä, että otetaan huomioon muiden kulttuurien edustajat. Yksikin huono viitsi riittää vieraannuttamaan mahdollisen pelaajan. (Adams 2010, 74.)

Toisin kun oletetaan, joidenkin kohdeyleisöjen välillä ei ole suuria eroja. Hyvin pitkään on ollut käsitys siitä, että naiset ja miehet ovat pelaajina hyvin erilaisia. Tutkimukset kuitenkin osoittavat, että miehet ja naiset pitävät hyvin paljolti samanlaisesta sisällöstä. Ero on niin pieni, että pienikin muutos sisältöön riittää tehdäkseen pelin kohdeyleisöstä kattavamman. (Adams 2010, 74.)

Suurin ero kahden ryhmän välillä on satunnaispelaajien ja niin kutsuttujen core-pelaajien välinen ero. ”Core” tulee sanasta hardcore. Core-pelaajille pelaaminen on harrastus, johon laitetaan paljon aikaa ja rahaa. He nauttivat haasteista. Satunnaispelaajat sen sijaan pelaavat nimensä mukaisesti satunnaisesti, eivät usein laita rahaa peleihin ja voivat vieraantua pelistä, jos siinä on liikaa haastetta. Tämä ero vaikuttaa suuresti siihen, minkälainen pelistä tulee, joten tiimin on päätettävä kenelle peli tehdään. On hyvin epätodennäköistä, että satunnaispelaaja nauttisi core-sisällöstä. (Adams 2010, 75.)

Kohdeyleisön lisäksi tärkeä osa konseptia on *lajityyppi* tai *genre*. Toisin kuin elokuvissa, pelin genre ei määrittele pelin tunnelmaa, vaan pelin pelattavuutta (gameplay). Pelin genre kertoo paljon siitä, miten peli toimii. Jos pelin genreksi valitaan FPS (First Person

Shooter), kaikki projektin osakkaat tietävät, miten hahmoa ohjataan pelissä ja mistä kuvakulmasta pelaaja tulee näkemään pelimaailman (Chandler 2014, 135). Perinteisiä genrejä ovat

- Seikkailupeli (Myst, Monkey Island)
- Toimintapeli (Devil May Cry, Street Fighter, Dota)
- Ammuntapeli (Half-Life, Call of Duty, Grand Theft Auto)
- Roolipeli (Fallout, Dragon Age, The Witcher, World of Warcraft)
- Strategiapeli (Civilization, StarCraft, Age of Empires, Total War)
- Urheilupeli (FIFA, NHL)

Lajityyppi voi olla myös *hybridi*, eli yhdistelmä kahta tai useampaa perinteistä genreä. Hyvä esimerkki hybridigenrestä on toimintaseikkailu (Adams 2010, 71), jonka tunnetuin edustaja on luultavasti Legend of Zelda –pelisarja. Tämä hybridigenre yhdistää toiminnan reaaliaikaisen taistelun seikkailupelien ongelmanratkonnan ja seikkailemisen kanssa.

Kun kohdeyleisö ja genre on päätetty, päätetään mikä on *kohdealusta*. Kohdealusta on enemmän kuin pelkkä laitteisto, millä peliä tullaan pelaamaan, sillä se vaikuttaa moneen asiaan, kuten

- Kontrolliskeemaan, eli miten pelaaja ohjaa peliä
- Teknisiin rajoitteisiin, jotka vaikuttavat esimerkiksi grafiikan laatuun
- Siihen, missä peliä pelataan, esimerkiksi olohuoneessa, työhuoneessa, julkisella paikalla (Schell 2015, 26)
- Verkkomoninpeliin

Koti on yksi yleisimmistä paikoista, missä pelataan. Melkein jokaisessa kodissa on nykyisin vähintään kannettava tai pöytäkone. Jos joku asukkaista pelaa aktiivisesti pelejä, usein kotoa myös löytyy jokin konsoli. PC:t ja konsolit kuitenkin eroavat toisistaan huomattavasti. Suurin ero on laitteen sijainnissa, eli sillä missä pelataan. Viime aikoina PC:t ovat jossain määrin päässeet myös olohuoneeseen, mutta konsolit ovat vieläkin olohuoneen pääasiallisia pelikoneita, kun taas PC:t ovat työhuoneessa. Olohuoneessa rentoudutaan, kun taas työhuoneessa tehdään keskittymistä vaativaa työtä (Schell 2015, 27-28).

Olohuoneessa pelikone, usein konsoli, käyttää päätelaitteenaan TV-näyttöä. TV on usein kaukana pelaajasta, joten on vaikeampi nähdä pieniä yksityiskohtia. Työhuoneessa on eri tilanne: päätelaite, PC-näyttö, on lähellä pelaajaa. Vaikka päätelaite onkin usein pienempi kuin TV, se pystyy TV:n tasoihin resoluutioihin, ja koska pelaaja on lähellä, hän pystyy

näkemään pienetkin yksityiskohdat. Tämä ero vaikuttaa useaan pelin ominaisuuteen, kuten esimerkiksi pelin grafiikan laatuun ja pelin käyttöliittymään. (Adams 2010, 78.)

Mobiililaitteet ovat viime vuosikymmenen aikana nousseet suosioon etenkin satunnaispelaajien keskuudessa. Kotona oleviin pelilaitteisiin, konsoleihin ja PC:hen nähden mobiililaitteet, älypuhelimet ja tabletit ovat helposti kuljetettavia (Adams 2010, 79). Kotona niitä käytetään lukunurkasta sohvaan (Schell 2015, 27), mutta erikoisia perinteisiin pelilaitteisiin nähden niistä tekee niiden kuljetettavuus. Ne ovat täten vallanneet julkiset paikat: jos julkisella paikalla pelataan jollakin laitteella, se on mobiililaitte.

Johtuen fyysisten painikkeiden puutteesta ja kosketusnäytöstä, pelit pitää suunnitella hyvin eri tavalla (Adams 2010, 79). Tämä kontrolliskeeman ero vaikuttaa suuresti esimerkiksi pelattavuuteen ja käyttöliittymään (Adams 2010, 79). Peliteollisuus on huomannut, että perinteiset peligenret eivät sovi mobiililaitteille. Toisin kuin kosketusnäyttö, joka on rajoite ja vahvuus samaan aikaan, mobiililaitteiden laitteisto ei kestä yli puolen tunnin pelisessioita. Tämä johtuu akun kestosta ja muista tekijöistä. Usein esimerkiksi älypuhelin lämpenee tukalan kuumaksi kädessä, kun mobiilipeliä on pelannut noin 15-30 minuuttia. Vaikka mobiililaitteiden prosessointiteho riittäisi 3D-grafiikoita varten, niitä ei ole suunniteltu pitkää rasiusta varten, mikä johtaa laitteen tukalaan lämpöön.

Käsi-konsolit ovat konsolien ja mobiililaitteiden välimaastosta. Niissä on perinteiset fyysiset painikkeet, mikä on mallinnettu konsoliohjaimista. Toisin kuin mobiililaitteet, ne on suunniteltu kestävämmään useamman tunnin pelisessioita. Tämän takia ne eivät esimerkiksi kuumene epämiellyttävästi kuumiksi pelejä pelattaessa. Nämä kaksi eroa tekevät niistä hyvin erilaisia mobiililaitteisiin nähden, vaikka niillä pelataankin hyvin samanlaisissa tiloissa, esimerkiksi vaikkapa lentokoneessa.

Tässä vaiheessa konseptointia tiedetään jo, mikä on pelin alustava konsepti, pelin kohdeyleisö, genre ja kohdealue. Tällöin päätetään miten pelillä tehdään rahaa, eli mikä on pelin *tulomalli* (revenue model). Peliteollisuudessa kolme yleisintä tulomallia ovat kiinteä hinta, kuukausimaksut ja mikrotransaktiot. Perinteisesti käytetty kiinteä hinta on yleisin tulomalli (Chandler 2014, 136), mutta nopeasti kasvavalla mobiilipelialalla yleisin on mikrotransaktiot, mitkä ovat alkaneet yleistyä myös perinteisten videopelien tuotannossa.

Kiinteä hinta on hyvin yksinkertainen: kuluttaja maksaa pelistä pelin hankinnan yhteydessä kiinteän hinnan, mikä antaa kuluttajalle, tulevalle pelaajalle, lisenssin (Chandler 2014, 136). Lisenssi antaa oikeuden pelin pelaamiseen, eli tuotetta saa käyttää henkilökohtaiseen viihdekäyttöön.

Kuukausimaksut ovat yleisin tulomalli massiivimoninpeleissä (Chandler 2014, 136). Amerikkalaisen pelitalon Blizzardin World of Warcraft –peli on varmaan tunnetuin esimerkki tästä tulomallista. Vaikka World of Warcraftilla on kiinteä hinta, jotta pääsee pelaamaan, pitää maksaa lisäksi myös kiinteä kuukausimaksu. Kuukausimaksulla rahoitetaan palvelimiin menevät kustannukset.

Mikrotransaktio-tulomallissa peli itse on usein ilmainen, niin sanotusti Free-To-Play (F2P), tai ilmaispelejä (Chandler 2014, 137). Usein kuitenkin joko suoraan oikealla rahalla tai oikealla rahalla ostettavalla virtuaalirahalla voi ostaa erilaisia pelin sisäisiä esineitä. Nämä esineet vaihtelevat visuaalisista esineistä, kuten pelaajahahmon vaatteista, eräänlaisiin boostereihin, jotka nopeuttavat pisteiden keräämistä tai tekevät hahmosta vahvemman. Ilmaispelejä nähdään enemmänkin palveluna kuin tuotteena, toisin kuin kiinteähintaiset pelit (Vankka 2014, 7).

Riippumatta siitä, minkä tulomallin tiimi valitsee, sillä pitäisi tehdä rahaa siinä määrin, että pelin kehittäminen on kannattavaa. Jos peli ei tee voittoa pelitalolle tai asiakkaalle, joka rahoittaa kehitystä, pelien kehittäminen ei jatkossa ole mahdollista. Sen takia on tärkeää, että tulomalli sopii yhteen aikaisemmin määriteltyjen konseptin ominaisuuksien kanssa. Esimerkiksi Core-pelaajat eivät usein ole mielissään mikrotransaktioista, mikä kannattaa ottaa huomioon tulomallia päätettäessä, jos he ovat kohdeyleisönä.

Konseptivaiheessa kannattaa suorittaa *SWOT-analyysi* (Strengths, Weaknesses, Opportunities, Threats). Suorittaminen kannattaa ajoittaa esitotannon ensimmäisille viikoille. Konseptin vahvuuksien, heikkouksien, mahdollisuuksien ja uhkien analysoiminen näin varhaisessa vaiheessa auttaa tiimiä havaitsemaan pelin vahvuudet ja ainutlaatuiset ominaisuudet, millä voidaan voittaa mahdollinen kilpailu. Samalla saadaan parempi kuva kilpailusta ja siitä, mitä uhkia kannattaa ottaa huomioon kehityksen aikana. (Chandler 2014, 138.)

SWOT-analyysia tehdessä tulee arvioida kilpailua ja verrata sitä pelin konseptin tämänhetkiseen muotoon. SWOT-analyysissa vahvuudet ja heikkoudet ovat sisäisiä tekijöitä, eli konseptin ominaisuuksia, joihin tiimi ja osalliset voivat vaikuttaa. Mahdollisuudet ja uhat taas ovat ulkoisia tekijöitä, kuten kilpaileva yritys julkaisemassa kilpailevan tuotteen samaan aikaan kun kehitystiimi julkaisee oman pelinsä. Ulkoisiin tekijöihin ei voi vaikuttaa, sillä tiimin jäsenet eivät voi millään tietää esimerkiksi kilpailevien yritysten aikatauluja. SWOT-analyysin esittäminen on tärkeää, kun hankitaan hyväksyntä asiakkaalta: sen esittäminen osana myyntipuhetta osoittaa ymmärrystä markkinoiden tämän hetkisestä tilanteesta. (Chandler 2014, 137-138.)

SWOT-analyysin lisäksi kannattaa suorittaa *kilpailuanalyysi* selvittämään tuleva ja tämän hetkinen kilpailu. Kilpailuanalyysi on markkinatietoisuuden lisäksi hyödyllinen, kun konseptia esitetään asiakkaalle hyväksyntää varten. Jos asiakas on julkaisija, yleinen kysymys asiakastapaamisessa on kuinka peli tulee pärjäämään kilpailua vastaan. Kuten SWOT-analyysissä, pelin ainutlaatuiset myyntivaltit ovat tärkeä selvityksen aihe. Kilpailijoista taas tulisi selvittää esimerkiksi hinta, sisällön määrä, journalistien antamat arvostelutulokset ja mitkä ovat kilpailevan tuotteen avainominaisuudet. (Chandler 2014, 139-141.)

Kun kaksi viikkoa on kulunut esituotannossa, kannattaa pitää asiakastapaaminen asiakkaan hyväksyntää varten. Asiakastapaamiseen mennessä tulisi olla valmiina alustava konsepti, kohdeyleisö, kohdealusta, genre, SWOT-analyysi ja mahdollisesti myös kilpailuanalyysi. Jos asiakkaalta hankkii hyväksynnän myöhemmin, kun töitä on tehty vielä enemmän, kaikki lisätyö on mennyt hukkaan, jos asiakas ei olekaan tyytyväinen projektin suuntaan. Asiakastapaamiseen kannattaa ottaa mukaan vain olennaiset henkilöt ja heistäkin vain ne, jotka ovat hyviä puhumaan. Jonkun läsnäolijan tulisi myös kirjata kaikki palaute mitä asiakkaalta saadaan. (Chandler 2014, 141.)

Kun asiakas on hyväksynyt konseptin, voidaan jatkaa konseptin kehittämistä. Teknisiä rajoitteita ei kannata vielä määritellä, koska tässä vaiheessa ei kannata sulkea pois mahdollisuuksia tekniikan osalta. Konseptointi jatkuu, koska tässä vaiheessa ei ole vielä lopullisesti määritelty seuraavia asioita:

- pelattavuutta (gameplay) ja pelimuotoja (game mechanics)
- miljöötä, maailmaa mihin peli sijoittuu
- tarinan tiivistelmää
- hahmoja
- konseptitaidetta
- äänimaailmaa
- yhtä tai useampaa prototyyppiä
- riskianalyysia

Kahden vuoden projektissa näiden osa-alueiden luomiseen tulee mennä yhdestä kahteen kuukauteen (Chandler 2014, 141). Joitakin osa-alueita ei viimeistellä aivan täysin: esimerkiksi tarinavetoisessa pelissä hahmojen ja miljöön suunnittelemiseen voi mennä useita kuukausia. Näitä parannellaan, kun siirrytään suunnitelman luontiin ja kun asiakas on antanut hyväksynnän konseptille kokonaisuudessaan.

Suurin osa suunnitteludokumentaatiosta menee *pelimuotojen ja pelattavuuden* kuvaamiseen. Tämä on hyvin laaja aihealue, joka kattaa

- mitä haasteita pelaaja tulee kohtaamaan
- miten pelaajaa tullaan palkitsemaan panostuksesta
- oppimiskäyrän, eli miten haastavaa pelin oppiminen tulee olemaan
- kontrolliskeeman, eli miten peliä ohjataan
- pelaajan toiminnot ja pelaajan rooli pelimaailmassa
- moninpelin

Pelattavuuden suunnittelussa kannattaa muistaa kohdeyleisö, sillä peliä tehdään heille eikä kehittäjiä varten. Suunnittelussa kannattaa muistaa pyrkiä kattamaan mahdollisimman monta kohdeyleisöä. Esimerkiksi oppimiskäyrää miettiessä kannattaa pyrkiä siihen, että peli on helppo oppia, mutta pelin mestaroiminen vaatii aikaa. Kontrolliskeeman tulisi olla helppo muistaa ja pelimuotoja tukeva. Esimerkiksi konsoliohjaimen olkapainikkeiden käyttö hahmon kääntämisessä ensimmäisen persoonan kameralla ei ole välttämättä luontevinta, etenkin jos on mahdollista käyttää oikeaa analogikeppiä. Jos pelissä tulee olemaan moninpeliominaisuus, se tulisi ottaa heti alussa huomioon. Moninpelituellinen peli on helpompi muuttaa yksinpeliksi kuin yksinpeli moninpeliksi (Gregory 2009, 45). Jos pelin pelattavuuden suunnittelu osoittautuu haastavaksi, pelattavuuden voi ottaa suoraan toisesta pelistä. Tätä kutsutaan *reskinnaamiseksi* (Portnow 2008).

Pelattavuudesta tehdään yksi tai useampi *prototyyppi*. Tässä vaiheessa saattaa olla liian aikaista tehdä digitaalisia prototyyppijä, joten prototyypit voivat olla paperimuodossa, ja ne voidaan tehdä käyttäen olemassa olevia lauta- ja korttipelejä. Myöhemmin, kun pelistä on tiedossa enemmän, tehdään digitaalisia prototyyppijä. Prototyypit, ovat ne missä tahansa muodossa, auttavat tunnistamaan ongelmia suunnittelussa pelattavuudessa tai pelimuodossa. Niillä voi myös tutkia mahdollisuuksia, hankkia ideoita ja jäsentää vaatimuksia. Vaikka prototyyppijä heitettäisiinkin pois, niistä opittu on erittäin tärkeää. (Chandler 2014, 146.)

Miljöö on yksi niistä ominaisuuksista, jonka yksityiskohtaisuus vaihtelee pelin genren ja tarinapainotteisuuden mukaan. Jos ollaan kehittämässä peliä, jossa tarina on olennainen osa pelikokemusta, miljöön tulee olla erityisen yksityiskohtainen. Miljöön suunnittelu on suuri tehtävä, ja siihen tulee panostaa vain, jos pelistä tehdään erityisen tarinapainotteinen (Adams 2010, 49). Miljööstä on vastuussa suunnittelija tai pääsuunnittelija, jos suunnittelijoita on useampi (Chandler 2014, 143). Miljöö on myös todella tärkeä tekijä pelin tunnelmassa ja visuaalisessa ulkonäössä (Adams 2010, 49).

Tarinan tiivistelmä on lyhyt kuvaus pelin tarinasta: mistä se alkaa ja miten se etenee. Tiivistelmä on hyvin samanlainen kuin kirjan takakansi. Tarina pitää pelaajan kiinnostuneena ja motivoi häntä jatkamaan pelaamista (Adams 2010, 50). Tarinan voi kertoa eri tavoilla, mutta yleensä se kerrotaan joko pelin aikana tai välivideoissa (cutscenes) (Adams 2010, 50). Tarina on hyvin kallis teettää, koska se on suuri osa-alue ja vaatii paljon aikaa toteuttaa. Pienen budjetin peleissä ei yleensä panosteta tarinaan (Adams 2010, 50).

Hahmot ovat olennaisia, jos pelissä tulee olemaan edes kevyt tarina. Hahmojen suunnittelussa konseptia varten tulee kirjoittaa pienet kuvaukset jokaisesta hahmosta, etenkin päähenkilöstä tai siitä hahmosta, millä pelaaja tulee pelaamaan. Pelaajan ohjaamaan hahmoon pitää pystyä samaistumaan helposti (Adams 2010, 48). Hahmojen ulkonäkökin pitää suunnitella, missä konseptitaiteella on suuri osa. Etenkin pelaajan hahmon tulee näyttää miellyttävältä, jos peli on kolmannesta persoonasta, koska silloin pelaajahahmo on jatkuvasti näytöllä (Adams 2010, 48). Hahmoista pitää myös määritellä mitä he voivat tehdä, mitä tunteita he tuntevat ja esimerkiksi mitkä ovat heidän mieltymyksensä (Adams 2010, 49).

Konseptitaide on yksi tärkeimmistä taiteilijoiden tehtävistä esituotannossa. Konseptitaide auttaa tiimiä visualisoimaan miltä hahmot, esineet ja maailman eri alueet näyttävät. Myös hahmojen kyvyt voidaan visualisoida kuvasarjoina. Yhdestä asiasta voidaan tehdä useampi kuva, kunnes tiimi on tyytyväinen siihen. Konseptitaide on myös erinomainen markkinointiväline. Ihmiset ostavat visuaalisesti miellyttävien näköisiä tuotteita, joten jos tiimillä on markkinointitiimi avustamassa markkinoinnissa, he pyytävätkin usein kehitystiimiltä konseptitaidekuvia mainontaa varten (Chandler 2014, 22). Konseptitaiteen luomiseen voi mennä useampi viikko (Chandler 2014, 144).

Äänimaailma on se, miltä peli kuulostaa. Se kattaa äänitehosteet ja mahdollisen musiikin. Tässä vaiheessa ei kuitenkaan suunnitella äänitehosteita ja musiikkia, vaan määritellään miltä ne tulevat kuulostamaan: mikä on niiden teema, minkälaisia ääniä pelimaailmassa tullaan kuulemaan ja mitä soittimia musiikissa tulee olemaan. Tässä vaiheessa myös päätetään tuleeko pelissä olemaan ääninäyttelyä (Chandler 2014, 145). Äänimaailman suunnitteluun voi osallistua koko tiimi, mutta siitä vastaa äänisuunnittelija (Chandler 2014, 33), jos sellainen on tiimissä, tai pääsuunnittelija. Musiikista vastaa säveltäjä, joka työskentelee suunnittelijan ja mahdollisesti äänisuunnittelijan kanssa (Chandler 2014, 33). Äänimaailmalla on suuri vaikutus pelin tunnelmaan ja tuntumaan. Esimerkiksi kauhupeleissä äänimaailmalla on suuri merkitys.

Konseptoinnin päätteeksi, ennen lopullista konseptin hyväksynnän hankkimista asiakkaalta, suoritetaan *riskianalyysi*. Tuottaja jäsentää riskit ja laittaa ne tärkeysjärjestykseen sen perusteella, kuinka paljon haittaa riskin toteutumisesta koituu ja kuinka todennäköistä se on, että riski toteutuu. Vaarallisin riski on sellainen, jolla on suuri mahdollisuus toteutua ja josta koituu erittäin paljon haittaa projektille (Chandler 2014, 148). Esimerkki suuresta riskistä on, jos pääinsinööri ei ole omistautunut projektille ja toinen yritys yrittää palkata häntä. Kannattaa kuitenkin muistaa, että joskus hyvin epätodennäköiset riskit voivat olla vaarallisempia kuin todennäköiset riskit (Chandler 2014, 148). Riskianalyysin tulosten pohjalta tuottaja ja tiimi pyrkivät estämään kaikista vaarallisimpien riskien toteutumisen tai vähentää niiden toteutumisen kielteisiä vaikutuksia (Chandler 2014, 151). Pääinsinöörin lähtemisen riskin voi mahdollisesti purkaa saamalla pääinsinöörin omistautumaan projektille, joko henkisesti tai sopimuksella, tai pyrkimällä siirtämään hänen osaamisensa toiseksi osaavimmalle insinöörille, josta tulisi riskin toteutuessa pääinsinööri. Tärkeää on, että projektille kriittinen osaaminen löytyy muultakin kuin pääinsinööriltä, jotta riski voidaan välttää.

Riskianalyysin jälkeen konsepti kokonaisuudessaan tulee esittää asiakkaalle hyväksyntää varten. Jos asiakas pitää konseptista, hän antaa luvan jatkaa projektia. Samoin kuin aiemmissa asiakastapaamisissa, mukaan otetaan ainoastaan olennaiset henkilöt. Joku läsnäolevista kirjaa asiakkaan palautteen.

3.1.2 Pelin vaatimukset

Kun asiakas on hyväksynyt konseptin, alkaa projektin vaatimusten määrittely. Kun aikaisemmin suunniteltiin korkealla tasolla, mitä peli tulee olemaan, nyt suunnitellaan, mitä pelin toteuttamiseksi tarvitaan konkreettisesti. Esimerkiksi kun aikaisemmin mietittiin äänimaailmaa, nyt suunnitellaan, mitä äänitehosteita tarvitaan ja miten paljon musiikkia tulee olemaan. Resurssitarpeiden mitoittamisen lisäksi määritellään käytettävä tekniikka ja työkalut. Osana vaatimusten määrittelyä kirjoitetaan dokumentaatiota vaatimuksista ja konseptista. Vaatimukset ovat osa lopullista pelisuunnitelmaa (game design document, GDD).

Tässä vaiheessa esituotantoa tulee määritellä pelin ominaisuuksia. Aivoriihet ovat siihen hyvä työkalu. Kaikkia ideoita ei kuitenkaan voi toteuttaa, koska aikaa, rahaa ja resursseja on rajallinen määrä. Tämän takia toteutettavat ominaisuudet tulee laittaa tärkeysjärjestykseen. Yleiset kolme prioriteettitasoa ovat (Chandler 2014, 156):

- Ensimmäinen taso (1st tier): ydinominaisuudet, pakolliset (must-have)
- Toinen taso (2nd tier): toissijaiset ominaisuudet, tukevat tai lisäävät arvoa ydinominaisuuksille (good to have)

- Kolmas taso (3rd tier): vapaaehtoiset ominaisuudet, jotka eivät välttämättä lisää arvoa muihin ominaisuuksiin (nice to have)

Aivoriihissä ominaisuudet kannattaa myös lajitella luokkiin. Luokkia on kolmea erilaista (Chandler 2014, 156):

- Prosessiin liittyvät ominaisuudet, joilla parannetaan kehitysprosessia, dokumentaatiota, ja pidetään koulutustilaisuuksia
- Tuotantoon liittyvät ominaisuudet, jotka ovat työkalu- ja tekniikkaehdotuksia
- Pelattavuuteen liittyvät ominaisuudet, jotka vaikuttavat suoraan pelaajan pelikokemukseen

Tiimin johto arvioi luokittelun jälkeen ominaisuuksien prioriteetit ja listaa ne.

Virstanpylväät (milestones) mittaavat kehityksen kulkua olemalla pieniä päämääriä kehityksen elinkaareissa. Koska virstanpylväiden välissä on vähemmän aikaa kuin esituotannon ja tuotannon päättymisen, ne tuntuvat saavutettavammilta. Sen sijaan, että tähdittäisiin heti alusta lähtien valmiiseen peliin, pyritään saavuttamaan julkaisua edeltävät virstanpylväät. Tämä pitää motivaation kunnossa kehitystiimissä pitämällä päämäärän saavutettavana. Lista virstanpylväistä päivittyy kehityksen läpi (Chandler 2014, 160). Yleisiä virstanpylväitä ovat

- ensimmäinen pelattava
- Alpha
- koodin lukitseminen (code freeze)
- Beeta
- julkaisukandidaatti (code release candidate)

Ensimmäinen pelattava on usein ensimmäinen suuri virstanpylväs kehityksessä, kun siirrytään tuotantoon. Ensimmäinen pelattava on nimensä mukaisesti ensimmäinen pelattava versio pelistä. Se perustuu prototyyppiin, mutta toisin kuin prototyyppi, sisältää pelin lopullista versiota vastaavan tason grafiikkaa ja pelattavuutta. Ensimmäinen pelattava on usein valmis vuotta ennen koodin lukitsemista kahden vuoden kehityssyklillä. (Chandler 2014, 159.)

Alpha on merkittävämpi virstanpylväs kuin ensimmäinen pelattava ja osoittaa projektin edistyneen pitkälle. Alpha -versiossa ydinominaisuudet ovat pelattavuuden osalta toteutettu ja sisältö vastaa noin 40–50% lopullisen tuotteen sisällön määrästä. Peli toimii debug-tilassa kohdealustalla. Ominaisuudet voivat vielä muuttua suurestikin pelitestauksen palautteen pohjalta. Alpha -versio saavutetaan yleensä kahden vuoden

kehityssyklillä kahdesta kahdeksaan kuukautta ensimmäisen pelattavan jälkeen.

(Chandler 2014, 159.)

Koodin lukitseminen tapahtuu, kun pelin koodi on valmis. Tämän jälkeen ominaisuuksia ei enään lisätä. Ominaisuuksia ei myöskään muuteta, jollei kyseessä ole bugi. Koodin lukitseminen tapahtuu usein kolmesta neljään kuukautta ennen julkaisua kahden vuoden kehityssyklillä. (Chandler 2014, 159.)

Beeta -versiossa molemmat pelin sisältö ja koodi ovat valmiita. Jäljellä on vain bugien korjausta ja laadunvarmistusta. Suurin osa tiimistä on siirtynyt muihin projekteihin ja jäljellä on vain joukko insinöörejä, joiden tehtävänä on korjata testauksessa ilmenevät bugit. (Chandler 2014, 159.)

Julkaisukandidaatti on sellainen versio pelistä, joka mahdollisesti pistetään levitykseen, kun peli julkaistaan. Julkaisukandidaatti valitaan, kun bugit ovat korjattu. Insinöörit lähettävät version, josta bugit ovat korjattu, testaajille, jotka testaavat sitä testaussuunnitelman mukaisesti. Julkaisukandidaatin tulee olla valmis testausta varten kolmesta neljään viikkoa ennen julkaisua, jotta testaajilla olisi riittävästi aikaa testata peliä. (Chandler 2014, 159.)

On joitakin ominaisuuksia, jotka ovat valmiita vasta Beeta -versiossa, eli kun sisältö on valmis. Hyvä esimerkki tällaisesta ominaisuudesta on tarina. Tämänlaisissa tapauksissa määritellään, miten paljon tarinan eri osista pitää olla valmiina mitäkin virstanpylvästä varten. (Chandler 2014, 161.)

Projektia varten pitää valita sopivaa *tekniikkaa*. Tekniikan soveltuvuus riippuu eri tekijöistä: kuinka paljon aikaa on kehittää oma talon sisäinen ratkaisu, onko pelitalossa osaamista ratkaisun kehittämiseen, mitä ominaisuuksia ratkaisulta halutaan ja kuinka laadukkaita ominaisuuksien pitää olla. Tekniikan hankinnassa pitää päättää halutaanko tehdä oma ratkaisu vai ostaa ulkopuolisen yrityksen ratkaisun lisenssi. Molemmissa on omat hyötynsä, sisäisesti kehitetyn ratkaisun kanssa ei tarvitse huolehtia lisenssistä ja tiimi tietää ratkaisun perin pohjaisesti, kun taas kolmannen osapuolen ratkaisun käyttäminen säästää kehitysaikaa ja parhaimmillaan mahdollistaa sen, että pelin työstämisen voi suoraan aloittaa ilman tekniikan kehittämistä. Arviointi tekniikasta annetaan tuottajalle, joka hyödyntää arviota aikataulun ja budjetin suunnittelussa. (Chandler 2014, 162.)

Tekniikan lisäksi pitää arvioida mitä *työkaluja* tarvitaan. Työkalujen valinnassa kannattaa miettiä, miten vähentää seisonta-aikaa (downtime). Jos joku joutuu odottamaan muiden

tuotoksia, kehitys hidastuu. Yhdellä henkilöllä ei saisi olla liikaa töitä. Resurssien saaminen peliin pitäisi olla mahdollisimman helppoa ja vaivatonta. Kannattaa miettiä, voiko prosessin automatisoida. Mitä vähemmän prosessi on automoitu, sitä todennäköisempää on, että virheitä tapahtuu prosessin suorittamisen aikana. Työkalujen pitää olla valmiita mahdollisimman pian, sillä niiden puuttuminen ei saa hidastaa kehitystä. Erityistä huolta tulee laittaa versionhallinnan valitsemiseen, jotta sisältöä voidaan hallinnoida ja seurata tehokkaammin. Jos taiteilijoille kehitetään jotain erityistyökaluja, työkalujen kehittäjien tulee auttaa taiteilijoita oppimaan työkalujen käytön. (Chandler 2014, 164.)

Dokumentaation tulee olla selkeää ja ymmärrettävää, sillä moni ihminen tiimin sisällä ja ulkopuolella on riippuvainen dokumentaation laadusta. Dokumenttien muotoilun tulee olla sellainen, että niitä on helppo lukea ja niistä saa selkeää tietoa projektista. Dokumenttia kirjoittaessa kannattaa miettiä, kenelle dokumenttia kirjoitetaan. Taiteilija ei ole välttämättä tietoinen teknisestä käsitteistöstä, joten työkalujen ohjeiden tulee olla ei-tekniellään kielellä. Dokumenttien tulee myös aina vastata nykytilannetta. Prototyypistä kannattaa pitää dokumentaatiota, koska se on helpompi formaatti kuin pelattava prototyyppi. Kaikilla ei ole aina aikaa pelata prototyyppiä saadakseen tietoa pelin ominaisuudesta. (Chandler 2014, 166-167.)

Suunnitteludokumentaatio tulee kirjoittaa pelattavuudelle ja pelimuodoille, joista päätettiin konseptoinnissa ja joita on mitoitettu eri vaatimuksia määriteltäessä. Näihin aiheisiin kuuluu käyttöliittymä, moninpeliominaisuudet, hahmot ja heidän taustansa, pisteytys ja pelaajan palkitseminen, tehtäväsuunnitelmat ja tehtävien kulku, kontrolliskeema tai -skeemat, pelaajan toiminnot ja toimenpiteet, tarina ja tarinan kulku, tekoäly ja sen haastavuus, esineet kuten aseet ja parannukset. Dokumentaation tulee olla sellaista, että insinööri, taiteilija, testaaja tai toinen suunnittelija voivat toteuttaa yksittäisen ominaisuuden alusta loppuun asti kysymättä neuvoa suunnittelijalta joka ominaisuuden suunnitteli. Kun ominaisuus on valmis, sitä tulisi pystyä kokeilemaan pelissä. Suunnittelijoiden kannattaa myös tehdä lista, mitä kaikkea dokumentaatiota ja prototyyppijä tarvitaan. (Chandler 2014, 167.)

Taidedokumentaatio koostuu tyylioppaasta (style guide), resurssilistasta (asset list) ja työkalujen ohjeista. Tyyliopas käsittelee pelin ulkonäköä ja visuaalista tyyliä. Se sisältää konseptitaidetta, väripaletteja ja esimerkkejä siitä, miltä peli tulee näyttämään (Chandler 2014, 168). Resurssilista on lista visuaalisista resursseista mitä pitää tehdä (Chandler 2014, 168). Listasta löytyy esimerkiksi mitä kaikkia 3D-malleja tarvitaan, jos peli käyttää 3D-grafiikoita. Työkaluohjeet ovat ohjeet taiteilijoiden käyttämiin työkaluihin, jotka ovat osittain insinöörien kirjoittamia (Chandler 2014, 168). Esimerkiksi jos projektissa taiteilijat

käyttävät animointiin Adobe Flash Professional –ohjelmistoa, animaatiot pitää tuoda ulos ohjelmistosta skripteillä. Nämä skriptit ovat yleensä insinöörien kirjoittamia, ja niiden käyttöön taiteilijat tarvitsevat ohjeet.

Tekninen dokumentaatio sisältää ohjelmointikäytännöt (programming practices, coding standards), teknisen suunnitelman ja insinöörien työkalujen ohjeet (Chandler 2014, 169). Ohjelmointikäytännöt kattavat nimeämiskäytännöt ja käytetyt tekniikat, kuten ohjelmointikielet. Tekninen suunnitelma taas sisältää prosessi-, luokka- ja muut mahdolliset kaaviot. Se kattaa kaiken siitä, miten tekninen toteutus suoritetaan (Chandler 2014, 169). Työkaluohjeet ovat kaikkia talon sisäisiä työkaluja varten, joita insinöörit käyttävät. Työkaluohjeiden työstäminen on suuri työ, jos tiimi käyttää omaa pelimoottoria, koska silloin kaikki komponentit pitää dokumentoida.

Lopuksi suoritetaan vielä uusi riskianalyysi. Tämän jälkeen haetaan hyväksyntä asiakkaalta jatkaa projektia. Kaiken ei tarvitse olla valmista asiakastapaamiseen mennessä. Dokumentteja kannattaa tarkistuttaa asiakkaalla sitä mukaan kun niitä valmistuu. (Chandler 2014, 170.)

3.1.3 Pelisuunnitelma

Suunnitelma, Game Design Document tai Game Plan, kertoo mitä pelin kehityksessä tehdään missäkin järjestyksessä. Se myös kertoo kuka tekee mitään ja milloin minkäkin pitää olla valmis (Chandler 2014, 173). Pelisuunnitelma on projektin suunnitelma eli minkä perusteella koko projekti suoritetaan esituotannon jälkeen, ja se on prototyypin lisäksi esituotannon tärkein toimitettava tuotos. Pelisuunnitelma määrittelee koko projektin mittakaavan, scopen, eli miten visio yhdistyy käytännön toteutuksen kanssa (Portnow 2008).

Pelisuunnitelman laatimiseksi tulee tietää, mitä projekti vaatii eli mitä määriteltiin aikaisemmin. Vaatimukset voivat kuitenkin tässä vaiheessa vielä muuttua. Suunnitelma itse voi muuttua vielä myöhemmin ja sitä tulisi päivittää. Ominaisuuksien valmistuminen tulee kirjata sitä mukaa kun ne valmistuvat, mikä muuttaa vaatimuksia. (Chandler 2014, 173.)

Kun suunnitelmaa laaditaan ja kun se muuttuu myöhemmissä vaiheissa, kannattaa ymmärtää projektiorganisaation resurssien riippuvuudet. Aikataulu, budjetti, henkilöstö ja sisällön määrä sekä laatu kaikki riippuvat toisistaan (Chandler 2014, 174). Muutos yhteen pakottaa muutoksien tekemisen muihin. Jos esimerkiksi budjettia joudutaan supistamaan, aikataulua joudutaan mahdollisesti lyhentämään ja sisältöä leikkaamaan muun muassa

sisällön laadun ylläpitämiseksi. Kaikki osa-alueet ovat niin suuria, että niitä on hankala ajatella samaan aikaan. Onkin hyvä aloittaa aikataulusta.

Aikataulun luonnissa kannattaa käyttää aikataulutukseen tarkoitettua ohjelmistoa, kuten vaikkapa Microsoft Projectia (Chandler 2014, 175). Niillä saa aikaan riittävän yksityiskohtaisia aikatauluja (Chandler 2014, 175). Myös, jos projektin aikana tulee muutoksia aikatauluun, muutokset on helppo kirjata. Ohjelmistot usein osaavat laskea aikataulun uudelleen, kun aikatauluun tehdään muutoksia (Chandler 2014, 187).

Kun aikatauluun lisää työtehtäviä, kannattaa muistaa, että valmis kenttä sisältää kentän konseptoinnin, suunnittelun, prototyypin kehittämisen, alustavan geometrian luonnin, teksturoinnin, yksityiskohtien lisäämisen ja bugien korjaamisen (Chandler 2014, 175). Tie ideasta käytännön luomukseksi on pitkä, ja vaatii useamman tiimin jäsenen työpanoksen. Työtehtävät kannattaakin pilkkoa pienemmiksi osiksi, jotka on helpompi jakaa tasaisesti tiimin jäsenten välillä.

Aikatauluun kannattaa sijoittaa useampi pieni virstanpylväs suurien virstanpylväiden lisäksi. Jos tehtävän määräaika on kaukana, tulee kiusaus lisätä ominaisuuksia. Pienillä virstanpylväillä voidaan siis helpottaa ominaisuuksien hallinnointia. (Chandler 2014, 175.)

Aikataulun laatimiseen kannattaa ottaa koko tiimi mukaan. Tällöin tiimin jäsenet tietävät miksi mikäkin määräaika on valittu ja ymmärtävät paremmin myöhästymisen seuraukset. Tuottaja, joka on vastuussa aikataulusta, voi myös kysyä suoraan työtehtävästä vastuussa olevalta kuinka kauan hänellä menee työn suorittamiseen. Kaikki tämä lisää tiimin jäsenten omistajuutta omasta työstään. (Chandler 2014, 175.)

Aikataulun laatimista varten kannattaa määritellä milloin mikäkin on valmis (Chandler 2014, 176). Esimerkiksi ominaisuus on yleensä valmis vasta, kun se on nähtävästi pelissä. Tuottajan tulisi tätä valmiin määritelmää päättäessä miettiä, mitkä ovat aivan pakollisia osia, eli mikä on minkäkin ominaisuuden MVP (minimum viable product). Kun tiimi tietää milloin heidän työnsä on valmista, he voivat tehdä työn valmistumiseen liittyvät merkinnät itse, mikä vähentää tuottajan työtä.

Tässä vaiheessa tulee tehdä alustava aikataulu, jossa on tärkeimmät päivämäärät ja ajankohdat (Chandler 2014, 177). Siihen tulee kirjata isojen virstanpylväiden, kuten Alpha- ja Beeta-vaiheiden määräajat ja määritellä milloin minkäkin näistä tulee olla valmis.

Toimitettavat tuotokset tulisi pilkkoa pienempiin osiin, tehtäviin, mitkä pitää toteuttaa jotta toimitettava tuotos on valmis. Prosessi, millä tämä suoritetaan on nimeltään *Work*

Breakdown Structure (WBS). Esimerkiksi yhtä kenttää varten tiimin johto määrittelee selkeät askeleet kentän valmistumiseen asti. Tämän jälkeen pidetään aivoriihi, missä tutkitaan, mitä kaikkea tarvitaan tehtävän suorittamiseksi. Tässä kannattaa käyttää preesensia ja aktiivista verbiä. Esimerkillinen tehtäväkuvaus on ”tee alustava kentän rakenne”. Kun tehtävät on avattu, ne lajitellaan osaston tai osaamisalueen mukaan ja laitetaan jotakuinkin kronologiseen järjestykseen. Taiteilijat työstävät taiteeseen liittyviä tehtäviä ja insinöörit tekniikkaan liittyviä tehtäviä. Esimerkiksi kentän alustava rakenne on kenttäsuunnittelijan tehtävä. Lopuksi tiimin johto päättää vielä kuinka paljon aikaa kuluu mihinkin tehtävään. Tässä kannattaa ottaa mukaan tiimin jäsen, jolle tehtävä annetaan, sillä hän tietää parhaiten kuinka paljon aikaa tehtävään kuluu. (Chandler 2014, 179-180.)

Kun toimitettavat ominaisuudet on pilkottu pienempiin osiin, ja niistä on määritelty tehtäviä, tuottajan tulisi laatia yksityiskohtaisempi aikataulu. Ensiksi tulisi tarkistaa tehtävät. Yhden tehtävän suorittaminen ei saisi kestää viittä päivää kauempaa. Ihanteellisesti tehtävän suoritus kestää noin pari päivää. Tehtävän omistaja, eli tiimin jäsen, jolle tehtävä on annettu, on itse vastuussa aika-arviosta. Jos tehtävän omistaja ei osaa arvioida tehtävän suorittamisen kestoa, osaamisalueen johtaja voi antaa oman aika-arvion kokemuksen pohjalta. Esimerkiksi teknisessä tehtävässä pääinsinööri voi antaa aika-arvion. Aika-arviota ei kannata yli- eikä aliarvioida. Aika-arviota ei tule kuitenkaan jättää tekemäksi, koska se voi antaa väärän kuvan aikataulusta. Kun arviot syötetään aikataulusohjelmistoon, se ei voi laskea projektin aika-arvioita oikein, jos yksittäisten tehtävien aika-arviot puuttuvat. (Chandler 2014, 180.)

Jos tehtävän suorittaminen menee yliajalle, voidaan tehtävän omistajan kanssa päättää leikataanko tehtävän sisältö vai lisätäänkö se peliin sellaisenaan. Ominaisuus voi olla riittävän hyvä lisättäväksi peliin, vaikka tehtävää ei suoritettaisikaan loppuun asti. (Chandler 2014, 182.)

Tuottajan tulisi yksityiskohtaisempaa aikataulua laatiessa muistaa lomat ja sairauspoissaolot. Aikatauluun kannattaa jättää tilaa poissaoloja varten. Tehtävien väliin kannattaa muutenkin jättää aikaa siltä varalta, että jonkin tehtävän suorittamiseen menee kauemmin kuin odotettiin. Tehtävien kesto kannattaa mitoittaa sen mukaan, että kahdeksan tunnin työpäivänä työntekijä tekee tehokkaasti töitä viidestä kuuteen tuntia. (Chandler 2014, 182.)

Aikataulun laadittaessa kannattaa muistaa myös osajien väliset riippuvuudet. Esimerkiksi jotta kenttäsuunnittelija saa rakennettua kentän, hänellä tulee olla insinöörin luoma koodi ja myös taiteilijoiden luomat 2D tai 3D -taideresurssit. Jos taiteilija tai insinööri myöhästyy tehtävässään, kenttäsuunnittelija ei pääse työstämään omaa tehtäväänsä. Koko

tuotantolinja voi pysähtyä yhden myöhästymisen takia. Tämän takia on erityisen tärkeää, että tehtävät jaetaan tasaisesti osaamisalueiden mukaan. Kenelläkään ei saisi olla liikaa töitä. (Chandler 2014, 183.)

Koko kehitystiimin tulee olla ajan tasalla tehtävien edistymisen osalta. Tiimille voi ilmoittaa eri tavoilla edistymisestä. Yksinkertaisen aikataulun voi tulostaa paperille ja laittaa oveen, mistä tiimin jäsenet näkevät sen (Chandler 2014, 187). Scrum –ketterää kehitysmenetelmää käyttäessä tehtävien edistyminen tulee ilmi Daily Scrumissa. Monimutkaisten aikataulujen kanssa tehtävienhallinnassa voi hyödyntää tehtävienhallintaohjelmistoa, kuten Atlassian Jiraa.

Jos joku on myöhästymässä tehtävänsä osalta, hänen tulee heti ilmoittaa siitä aikataulua pitävälle taholle. Parhaimmassa tapauksessa tieto myöhästymisestä tulee sen verran ajoissa, että sen osalta ehditään luoda varasuunnitelma. Jos taiteilija on myöhästymässä tehtävänsä kanssa, voidaan tehtävää työstämään laittaa toinen taiteilija, jolloin tehtävä voidaan sittenkin suorittaa määräaikaan mennessä. (Chandler 2014, 187-188.)

Kun aikataulu on tiedossa, tulee miettiä keitä tiimi tarvitsee projektin tekemistä varten. *Henkilöstöä* suunnitellessa kannattaa muistaa henkilöstön riippuvuus aikataulusta ja rahoituksesta. Enemmän työntekijöitä tarkoittaa enemmän palkkojen maksamista. Jos ei ole rahaa, projektin rajausta joudutaan pienentämään vähemmän kunnianhimoiseksi. Pienempi budjetti näkyy esimerkiksi tiimin koossa ja aikataulussa. Itse henkilöstösuunnitelma perustuu aikataulun ja rahoituksen lisäksi siihen, mitä tarvitaan. Projekti tarvitsee useamman taiteilijan, jos peli vaatii 10 hahmoa ja viisi kenttää. Jos on uutta tekniikkaa, tarvitaan useampi insinööri sen toteuttamista varten. (Chandler 2014, 188.)

Henkilöstö voi muuttua läpi projektin, etenkin isoissa projekteissa. Ainoastaan esituotannon pieni aloitustiimi on mukana koko projektin läpi. Tuotannossa työntekijöitä tulee sisään ja menee ulos sitä mukaan, kun heitä tarvitaan. Beeta-vaiheessa ainoastaan pieni joukko insinöörejä on jäljellä, jotka korjaavat bugeja saadakseen pelin julkaisukelpoiseen kuntoon. Muutokset henkilöstöön kuvastuvat aikataulussa. (Chandler 2014, 189.)

Aikataulun ja henkilöstön suunnittelun jälkeen suunnitellaan *budjetti*. Projektin budjetin tulee olla kohtuullinen pelin laatuun ja aikatauluun nähden. Sen tulee ottaa myös huomioon myös se, että pelillä pitää tehdä voittoa. Etenkin asiakas haluaa, että tuotantokustannukset ovat kohtuulliset. Kohtuullisuuden arvioimiseksi asiakas tekee mahdollisesti tuloslaskelman, eli vertailee kustannuksia ennustettuihin myyntilukuihin.

Alustavan budjetin suunnittelun jälkeen tiimin pitää olla valmis muuttamaan sitä. Muut suunnitelmat voivat vielä vaikuttaa siihen ja projektin aikana tulee varmasti odottamattomia kustannuksia. Siinä tilanteessa, kun odottamattomia kustannuksia tulee, tiimin tulee pysyä rauhallisena ja projektiorganisaation resurssit pitää sijoittaa uudelleen. (Chandler 2014, 189-190.)

Kun budjettia, aikataulua ja henkilöstöä suunnitellaan, kannattaa miettiä *ulkoistamista*. Ulkoistamisessa on omat hyötynsä ja haittansa, mutta se on hyvä tapa mahdollisesti säästää aikaa ja rahaa. Monen asian voi ulkoistaa ilman, että sillä on vaikutusta aikatauluun, kuten (Chandler 2014, 196)

- välikohtaukset, videot
- liikkeenkaappauksen (motion capture, mocap)
- ääninäyttelyn
- musiikin
- ääniefektit
- kirjoituksen, tarina
- lokalisaation

Insinöörien töitä ei kannata ulkoistaa, koska kaikki ovat riippuvaisia insinöörien tekniikasta. Versionhallinta, bugien korjaus ja koodin testaus ovat hankalia tehdä, jos ei ole paikan päällä. Kannattaa myös muistaa, että pelin koodi on salassa pidettävää ja ulkoistettaessa sen muokkaamis- ja lukuoikeudet pitää antaa ulkopuoliselle. (Chandler 2014, 196.)

Ulkoistettaessa toimittajalle tulee ilmoittaa määräaika ja yhteyshenkilö. Yhteyshenkilö on tiimin jäsen, johon toimittaja ottaa yhteyttä tilanteissa, joissa se on tarpeellista. Toimittajalle kannattaa myös kertoa mahdollisimman paljon pelistä. Asiat kuten tarina, miljö, konseptitaide sekä pelattavat demot antavat hyvän kuvan pelistä ja auttavat esimerkiksi säveltäjää säveltämään peliin tunnelmaltaan sopivaa musiikkia. (Chandler 2014, 196.)

Ulkoistamisen seurauksena säästyvä aika voidaan käyttää bugien korjaamiseen ja laadun parantamiseen. Ulkoistaminen kuitenkin vähentää aikataulun joustavuutta. Määräaikojen muuttaminen vaatii yhteydenoton toimittajaan, usein sähköpostilla. Toinen ongelma, joka koituu ulkoistamisesta, on se, että kehitystiimi on riippuvainen siitä, saako toimittaja ulkoistetun osan valmiiksi määräaikaan mennessä. Sama pätee myös toisin päin: toimittaja on usein riippuvainen kehitystiimin aikaansaannoksista. Toimittajaksi kannattaa valita jokin luotettava organisaatio. (Chandler 2014, 196-197.)

Väliohjelmistoilla (middleware) voidaan säästää huomattavasti insinöörien aikaa. Sen sijaan, että insinöörien pitäisi kehittää ratkaisut animaatiolle, fysiikalle, renderöinnille ja verkko-ominaisuuksille, niiden osalta voidaan lisensoida kolmannen osapuolen kehittämä ratkaisu (Chandler 2014, 198). Esimerkki tästä on Nvidia -grafiikkateknologiyhtiön fysiikkamoottori PhysX, joka on yleinen fysiikkaväliohjelmisto nykyajan pelimoottoreissa, kuten Unity3D:ssä.

Nykyään on yleistä hankkia valmiiksi kehitetty pelimoottori, kuten Unity3D tai Unreal Engine. Aikaisemmin valmiit pelimoottorit piti muokata yhteensopivaksi pelattavuutta ja pelimuotoja varten (Gregory 2009, 12). Esimerkiksi tappelupelejä varten jouduttiin kehittämään oma pelimoottori, joka pystyi tunnistamaan monimutkaisia syöteyhdistelmiä (Gregory 2009, 17). Nykyään Unity3D:n ja Unreal Engine:n kaltaiset pelimoottorit ovat hyvin monipuolisia ja soveltuvat melkein kaikkeen paitsi simulaatioihin. Näissä valmiissa pelimoottoreissa on usein laadukkaita väliohjelmistoja komponentteina. Esimerkiksi Unreal Engine 3:ssa löytyy sisäänrakennettuna Autodesk GameWare Pathfinding -reitinhakuratkaisu (Autodesk 2015).

Aluksi oppimiskäyrä voi hidastaa kehitystä, kun insinöörit joutuvat opettelemaan uuden tekniikan, mutta pitkällä aikavälillä väliohjelmistot säästävät aikaa. Niitä suurempi haitta on lisenssikustannukset. Kolmannen osapuolen tarjoama tekninen tuki on kuitenkin usein laadukasta, ja hinnat ovat hyvin kilpailukykyisiä. (Chandler 2014, 198.)

Samaan aikaan, kun tuottaja suunnittelee budjettia, aikataulua ja henkilöstöä, insinöörit ja kenttäsuunnittelijat työstävät *prototyyppejä*. Viimeisessä asiakastapaamisessa asiakkaalle tulee esittää digitaalinen pelattava prototyyppi. Melkein jokaisesta pelistä on tehty useampi prototyyppi esituotannon aikana (Adams 2010, 50). Esituotannon digitaalisessa prototyypissä tulee olla vähintään kaikki aikaisemmin prototyypeillä testatut pääominaisuudet pelattavassa muodossa. Grafiikan ei tarvitse olla lopullista, eikä sen tulekaan olla (Adams 2010, 50). Tiimin kannattaa pitää mielessä, että todennäköisesti yhtään digitaalisen prototyypin sisällöstä ei välttämättä päädy lopulliseen peliin (Adams 2010, 50). Prototyyppi todistaa ideoitten toimivuuden käytännössä (proof of concept, POC).

Kun esituotannon tarvittava dokumentaatio sekä pelattava prototyyppi ovat valmiita, voidaan pitää *viimeinen esituotannon asiakastapaaminen*. Tässä asiakastapaamisessa asiakas päättää, aikooko hän rahoittaa pelin kehityksen eli saadaanko budjetti projektia varten. Asiakkaalle esitetään tässä vaiheessa valmiina oleva dokumentaatio ja pelattava prototyyppi, mistä asiakas saa käsityksen siitä, millainen lopullinen peli mahdollisesti tulee

olemaan. Jos asiakas hyväksyy esituotannon tulokset ja antaa rahoituksen, voidaan siirtyä tuotantoon. (Chandler 2014, 9.)

3.2 Tuotanto

Esituotannon päättyessä tiimillä on hyvä käsitys pelistä, jota ollaan tekemässä. Kehitystiimi tietää kuka tekee mitään ja milloin. Esituotannosta tuotantoon siirtyminen ei kuitenkaan ole selvärajainen, vaan usein vaiheittaista. Sitä mukaa, kun suunnitelmat valmistuvat, tiimin jäsenet siirtyvät työstämään pelin sisältöä. Taiteilijat esimerkiksi voivat olla valmiita jonkin pelihahmon kanssa, mutta ovat samalla suunnittelemassa jotakin toista.

Toinen asia joka tekee epäselväksi milloin projekti on esituotannossa ja milloin tuotannossa on, että osa sisällöstä pitää olla valmis esituotannon viimeiseen asiakastapaamiseen mennessä. Koska pelin markkinointi pitää aloittaa mahdollisimman aikaisin, osa pelin sisällön resursseista ja konseptitaiteesta pitää olla valmis markkinointia varten jo esituotannossa. (Chandler 2014, 9.)

Tuotannon alussa tuottaja esittää lopullisen pelisuunnitelman koko kehitystiimille. Tuottajalla on velvollisuus pitää suunnitelma ajantasalla läpi projektin projektin johdon kanssa. Kaikki mahdolliset muutokset, kuten virstanpylväiden saavuttamiset ja visuaalisen ilmeen muutokset, tulee kirjata suunnitelmaan. Jos muutoksia ei kirjata ajoissa, on mahdollista, että muutoksia ei toteuteta. (Chandler 2014, 10.)

Lista toteutettavista ominaisuuksista tulee pitää mahdollisimman lyhyenä. On parempi julkaista peli, jossa on vähemmän ominaisuuksia, kuin julkaista huomattavasti myöhemmin peli, jonka kehitys on ylittänyt rahoituksen ja määräajat ominaisuuksien määrän takia. Kehitystiimin tuleekin varoa, ettei tuotannon edistyessä ryhdytä keksimään lisää ominaisuuksia, jotka kasvattavat projektin mittakaavaa. (Chandler 2014, 10.)

Tuotannon edistyksen seuraamiseen riittää Excel -tiedosto. Edistyksen seuraamiseen ei kannata käyttää ohjelmistoa, joka ei ole tuttu, koska edistyksen tilaa tullaan päivittämään usein. Edistyksen kannalta on olennaista, että on tiedossa, milloin ominaisuus on valmis. Näin tiedetään, milloin ominaisuudet ovat valmiita. Tuottajan ja kehitystiimin tulisikin mahdollisimman pian päättää, mikä tarkoittaa, että ominaisuus on valmis (definition of done tai exit criteria). (Chandler 2014, 11.)

Tuotannossa suunnittelijat aloittavat toteuttamaan suunnitteludokumentaation sisältöä käytäntöön ja parantelemaan niitä osuuksia, jotka on jo toteutettu esimerkiksi

prototyyppiin. Ominaisuudet pelattavuudesta kenttien rakenteeseen muuttuvat jatkuvasti Beeta -virstanpylvääseen asti ja tärkeä osa tätä *suunnittelusykliä* on suunnittelijoiden pelitestausta ja heidän saama palaute testaajilta. Suunnittelijat hiovat ja parantelevat pelin toteutusta kunnes peli on aivan täydellinen. (Chandler 2014, 207.)

Esituotannossa suunnittelijat määrittävät pelin pelattavuutta, pelimuotoja, tarinaa, hahmoja, miljöötä, kenttiä ja äänimaailmaa. Osa näistä on testattu esituotannossa, jonka jälkeen niitä on mahdollisesti paranneltu ja lisätty lopulliseen prototyyppiin. Tämä suunnittelusykli jatkuu ja toistuu tuotannossa. Suunnittelusykli alkaa siis ominaisuuden toteuttamisella, jonka jälkeen ominaisuutta testataan ja parannellaan palautteen pohjalta, kunnes ominaisuus on täydellinen (Chandler 2014, 207). Koska suurin osa pelattavuudesta toteutetaan ensimmäisiin virstanpylväisiin, ensimmäiseen pelattavaan ja Alphaan, suurin osa suunnittelijoiden ajasta menee ominaisuuksien parantamiseen (Chandler 2014, 207). Esimerkiksi jokin kenttä voi olla jo valmis, eli se toimii ja on täysin pelattava, mutta kenttäsuunnittelija jatkaa sen hiomista kehityksen läpi. Samalla, kun taidetta ja koodia tuotetaan, ne tuodaan yhteen luomaan uudet ominaisuudet.

Esituotannossa taiteilijat päättävät esineistä, pelimaailmasta ja hahmoista suunnittelijoiden kanssa. Tuotannossa konseptitaiteen ja suunnitteludokumentaation pohjalta aloitetaan luomaan itse pelin sisäisiä resursseja. 2D-peleissä nämä ovat 2D-kuvia hahmoista, rakennuksista, maisemista ja esineistä. 2D-kuvat, niin kutsutut spritet, kuvaavat myös animaatiot ja erikoisefektit. 3D-pelejä varten tehdään 3D-malleja ja niiden tekstuurit sekä mahdollisesti animaatiot.

Taiteilijoille tuotanto on hyvin samanlainen kuin suunnittelijoille. Tarvittavat taideresurssit ovat usein valmiita ensimmäisiin virstanpylväisiin mennessä, joten iso osa tuotannosta menee aikaisemmin tuotettujen resurssien parantamiseen (Chandler 2014, 208). Esimerkiksi aina on jokin animaatio mitä voi parannella. Usein valmiit resurssit eivät vastaa laatuvaatimuksia, joten parannukset ovat pakollisia. Taidesykli, joka on samanlainen palaute- ja parannusprosessi kuin suunnittelusykli, toteutetaan useamman kerran ja tehdään tärkeysjärjestyksessä, jossa tärkeimmille resursseille annetaan palautetta ja niitä parannellaan ensiksi (Chandler 2014, 208). Esimerkiksi pelaajahahmo on tärkeämpi kuin jokin pelin sisäinen rakennus, joten sen parantamiseen laitetaan enemmän aikaa.

Insinöörit toteuttavat tuotannossa loput suunnitteludokumentaation määrittelemistä ominaisuuksista (Chandler 2014, 209), joita ei ole vielä toteutettu viimeiseen prototyyppiin. Insinöörien tehtävänä on myös korjata bugeja (Chandler 2014, 209) ja parantaa koodin laatua, olisi se sitten optimointia tai vain sieventämistä. Kuten taiteilijoilla ja

suunnittelijoilla, suurin osa insinöörien ajasta menee ominaisuuksien korjaamiseen tai parantamiseen. Koodin lukitsemiseen mennessä pelin koodin tulisi olla valmis, minkä jälkeen insinöörit vain korjaavat bugeja.

Pelin kehityksen aikana on hyvin tärkeää, että taide-, suunnittelu- ja tekniikkaosastot ovat yhteydessä toisiinsa ja tulevat toimeen keskenään (Chandler 2014, 210).

Väärinymmärrykset ovat yleisiä organisaatiossa, jossa osastot eivät koskaan ota yhteyttä toisiinsa (Chandler 2014, 210).

Osastojen välinen viestintä on tärkeää palautteen saamisessa osastojen tuotantokierroissa (Chandler 2014, 210). Vaikka osa palautteesta tulee toisilta osaston jäseniltä, esimerkiksi taiteilija ei voi kunnolla tietää onko taideresurssi yhteensopiva pelin suunnitelman kanssa, jollei hän saa palautetta suunnittelijoilta. Taiteilija ei myöskään voi kunnolla tietää toimiiko taideresurssi kohdealustan teknisten rajoitteiden puitteissa jollei taiteilija saa palautetta insinööreiltä.

Palautteen tulee noudattaa sopivan viestinnän sääntöjä, eli esimerkiksi kritiikin tulee olla rakentavaa. Kun palautetta annetaan, kannattaakin aina antaa ehdotuksia siitä, miten ongelma ratkaistaan. Jos taiteilijan 3D-mallissa on liikaa kulmia, mikä tekee siitä raskaan prosessoida laitteelle, insinöörin kannattaa kertoa miksi kulmia on liikaa ja mistä kulmia voisi vähentää.

Yhteistyöhön myös kuuluu muiden tiedottaminen siitä, milloin tehtävä on valmistumassa (Chandler 2014, 210). Tyhjäkäynnin vähentämiseksi tehtävien edistyminen kannattaa viestittää muille ajoissa ja mahdollisimman järjestäytyneellä sekä selkeällä tavalla (Chandler 2014, 210). Esimerkiksi Scrum -ketteräkehityskehikon päivittäinen kokous, Daily Scrum, on hyvä ratkaisu tähän.

3.3 Testaus

Vaikka testaajat ovatkin tekemisissä projektin kanssa jo esituotannossa, suurin osa testauksesta suoritetaan tuotannossa. Kun esituotannossa testaajat ovat vastuussa prototyyppien viihdearvon arvioimisesta ja ovat mukana vain osa-aikaisesti, tuotannossa he siirtyvät projektin pariin täyspäiväisesti. (Chandler 2014, 237.)

Ison budjetin projekteissa Alpha-vaiheen aikana muodostetaan vakituinen testaajatiimi. Alpha-virstanpylvästä lähtien he ovat koko loppuprojektin ajan mukana testaajina. Koodin lukitsemiseen mennessä testaajatiimi on saavuttanut maksimikoon. (Chandler 2014, 237.)

Kahden vuoden projektissa testaajat viettävät tuotannon loppupuolella viidestä kahdeksaan kuukautta testaten peliä. Pelistä riippumatta pelaamisesta tulee rasittavaa, kun sitä joutuu tekemään päivät pitkät. Peliprojekteissa on tapana allokoida liian vähän aikaa testaukselle, joten usein testaajat joutuvat tekemään ylitöitä sekä tulemaan töihin viikonloppuisin. Silti kaikkea ei todennäköisesti ehditä testaamaan. Testaukselle varattu aika on erityisen lyhyt etenkin silloin kun muu tiimi on myöhässä omien tehtäviensä osalta. (Chandler 2014, 235.)

Koska testaukselle on usein liian vähän aikaa, on tärkeää, että testaukselle on aikataulu. Testausaikataulun tulee perustua tuotantoaikatauluun ja siinä pitää saada hyvä käsitys riippuvaisuuksista, eli milloin mikäkin ominaisuus voidaan testata. Tuottajan tulee aikatauluttaa joitakin päiviä jokaisen virstanpylvään testaamiseksi, jotta voidaan tarkistaa vastaako tuotettu virstanpylväälle asetettuja vaatimuksia. (Chandler 2014, 236.)

Testausaikataulun laatimisessa pitää muistaa (Chandler 2014, 236)

- pelitestaus (play testing)
- pelattava esittelyversio (demo)
- markkinointiversiot
- julkaisukandidaatit (code release candidates)

Pelitestauksessa testataan peli viihdyttävyyttä sen sijaan, että etsittäisiin ohjelmistovirheitä tai bugeja. Pelitestaukseen kannattaakin sen takia tuoda ulkopuolelta uusia testaajia, jotka eivät ole jo kyllästyneet peliin ja jotka voivat nähdä pelin uudelta kannalta. (Chandler 2014, 236.)

Pelattava esittelyversio on versio pelistä, mikä esitetään messutapahtumissa ja muissa journalistien tapahtumissa. Tämä on markkinoinnin kannalta tärkeä versio pelistä.

Markkinointiversiot pelistä ovat journalisteille lähetettäviä esikatsausversioita. Vaikka journalistit odottaisivatkin bugeja, kriittiset bugit kannattaa korjata version luomiseen mennessä. (Chandler 2014, 236.)

Olennaista testausvaiheessa on palaute-korjaus -sykli, missä testaajat testaavat peliä ja raportoivat löytämänsä ongelmat bugiraporteilla insinööreille, jotka korjaavat ongelmat (Chandler 2014, 13). Ongelmien korjaamisen jälkeen insinöörit rakentavat uuden version testaajille testattavaksi (Chandler 2014, 13). Viikon aikana voidaan rakentaa useampi versio pelistä testaajille testattavaksi.

Bugit tulisi laittaa tärkeysjärjestykseen haitan mukaan, kuten pelin ominaisuudet esituotannossa. Bugit, mitkä rikkovat pelin keskeistä pelattavuutta tai estävät etenemisen, pelissä ovat erittäin haitallisia, joten niiden korjaaminen on tärkeintä.

3.4 Jälkituotanto tai projektin päättäminen

Jälkituotannossa arvioidaan menestymisiä ja epäonnistumisia, arkistoidaan tehdyt tuotokset dokumenteista itse peliin ja siirretään seuraavaan projektiin sisältöä, jota voidaan hyödyntää. (Chandler 2014, 15.)

Dokumentit, taiteilijoiden tuotokset ja pelikoodi paketoidaan yhdeksi kokonaisuudeksi, ja säilötään projektin portfoliona (Chandler 2014, 15.). Jos tulevaisuudessa tulee jokin ongelma, joka on ratkaistu joskus aikaisemmin, ratkaisu löytyy arkistosta (Chandler 2014, 15.). Arkistoinnista on myös hyötyä silloin, kun haluaa julkaista pelin uudelleen vaikkapa paremmilla grafiikoilla. Silloin, sen sijaan, että kehittää kaiken uudelleen, tarvitaan vain pelin julkaisuversio ja siihen lisätään parannellut taideresurssit.

Yleensä pelikoodi on se, mitä siirretään seuraaviin projekteihin, mutta jos työn alla on jatko-osa aikaisemmalle pelille, voi taidettakin kierrättää. (Chandler 2014, 15.)

Lopuksi vielä kirjoitetaan retrospektiivi. Retrospektiivejä voi olla useampia, riippuen tiimin koosta ja työn määrästä. Päämääränä kuitenkin on, että tiimi oppii onnistumisistaan ja epäonnistumisistaan ja voi tulevaisuudessa hyödyntää tätä tietoa muissa projekteissa. Retrospektiiveistä on myös hyötyä muissa projektin vaiheissa, ja niitä kannattaakin tehdä jokaisen virstanpylvään yhteydessä (Chandler 2014, 15).

Retrospektiivissä kannattaa miettiä näitä kysymyksiä (Chandler 2014, 258):

- Saavutettiin pelin päämäärä? Mitkä kaikki alkuperäisistä projektin päämääristä saavutettiin ja ovat pelin lopullisessa versiossa?
- Olivatko projektin aikataulu, resurssit, ominaisuudet ja laatuvaatimukset realistisia pelin päämääriin verrattuna? Missä asioissa nämä olivat mitoitettu oikein ja missä asioissa ei?
- Mikä meni hyvin ja mikä meni huonosti? Mikä toimi hyvin ja mikä ei?
- Mitä tärkeitä asioita opittiin tässä projektissa?

4 Tiimi

4.1 Työroolit

Koska pelit ovat viihdettä, toisin kuin perinteiset ohjelmistot, niiden kehittäminen vaatii muitakin kuin insinöörejä. Insinöörienkin tulisi olla luovia, mutta luovan osan edustajia kehitystiimissä ovat suunnittelijat ja taiteilijat. Heitä valvomassa on tuottaja, joka on peliprojektien projektipäällikkö. Hän ei välttämättä pääätä pelin sisällöstä, mutta hän on olennainen, kun halutaan tehdä aikataulu, budjetti tai henkilöstösuunnitelma ja noudattaa sitä. Hän on pääsuunnittelijan kanssa se, joka pitää projektin kasassa projektin elinkaaren läpi. Yhdessä nämä eri alojen osaajat toteuttavat käytännössä pelin vision.

4.1.1 Insinööri

Insinööri tai koodaaja on tiimin tekninen osaaja. Hän on olennainen osa tiimiä projektin elinkaaren läpi. Kaikki, mitä pelistä löytää, on olemassa insinöörin ansiosta. Hän mahdollistaa pelin olemassaolon. (Chandler 2014, 25.)

Insinöörit toteuttavat projektin työkalut, joita kenttäsuunnittelijat käyttävät kenttien luomiseen, joita taiteilijat käyttävät saamaan taideresurssit peliin ja joita insinöörit käyttävät pelin logiikan ohjelmoimiseen. Insinöörit vastaavat myös uudesta tekniikasta. Jos tekniikka tulee kolmannelta osapuolelta, he integroivat sen projektin työkaluihin. Jos tekniikkaa ei hankita kolmannelta osapuolelta, he kehittävät sen itse. (Gregory 2009, 5.)

Insinöörit voivat olla yleisosaajia, jotka voi laittaa työstämään mitä vain tehtävää. Yleensä insinöörit kuitenkin erikoistuvat johonkin osa-alueeseen. Pienissä tiimeissä pari insinööriä on vastuussa kaikesta teknisestä toteutuksesta, mutta suurissa organisaatioissa, jokaiselle osa-alueelle on erillinen asiantuntija. Silloin insinööri on vastuussa joko tekoälystä, fysiikasta, renderöinnistä tai jostain muusta pelin tai pelimoottorin ominaisuudesta (Chandler 2014, 26). Tiimin johtavan insinöörin tulee osata myös projektinhallintaa (Chandler 2014, 26).

Insinöörillä on usein tietojenkäsittelyn tutkinto, mutta hän voi olla myös itsekoulutettu koodaaja (Chandler 2014, 27). Tärkein taito insinöörillä on ohjelmointitaito (Chandler 2014, 27). Koska pelit vaativat suorituskyykyä, tärkein ohjelmointikieli osata on C++. Suurin osa työkaluista ja tekniikasta on ohjelmoitu C++:lla. Insinöörin tulisi pysyä ajan hermolla ja jatkuvasti päivittää osaamistaan (Chandler 2014, 27).

Pelejä koodatessa koodaajan tulee ymmärtää ero pelien ja perinteisen ohjelmiston välillä: pelit ovat viihdettä, joten koodin päämäärä on luoda viihdyttävä pelikokemus.

Käytännöllisyys on toissijaista, joskin käytännöllisyys ja viihdearvo eivät ole ristiriidassa keskenään. Pelikoodaajan tulee myös pystyä tulemaan toimeen tiimin taiteellisten jäsenten kanssa. (Chandler 2014, 25.)

Projektiorganisaation johtava insinööri on yleensä useamman vuoden työskennellyt pelien parissa. Hänellä tulee olla laajan teknisen osaamisen lisäksi hyvät projektihallinnolliset taidot. Suurissa projekteissa hän ei välttämättä koodaa lainkaan, mutta hallinnoi kaikkia muita insinöörejä, päättää tekniikasta, työkaluista ja ohjelmointikäytännöistä sekä auttaa tuottajaa laatimaan aikataulun ja budjetin (Chandler 2014, 26). Hän avustaa insinöörejä päivittäisessä työssä (Chandler 2014, 26). Hän on johtavan suunnittelijan ja taiteilijan kanssa osa ydintiimiä, joka on vastuussa pelin esituotannosta.

4.1.2 Taiteilija

Taiteilija on vastuussa kaikesta pelin visuaalisesta sisällöstä. Taiteilija on erityisen tärkeä myös esimerkiksi markkinoinnin kannalta. Konseptitaide on tärkeä työkalu pelin markkinoinnissa, koska se on varhaisin asia mitä pelistä voidaan näyttää. Konseptitaide usein tehdään erittäin laadukkaaksi vakuuttamaan asiakas ja kohdeyleisö. (Chandler 2014, 22.)

Taiteilija työskentelee koodaajan kanssa toteuttaessaan suunnittelijan visiota pelistä. Koodaaja tietää tekniset rajat eli mitä voidaan ja ei voida tehdä, sekä ketkä ovat vastuussa teknisestä toteutuksesta (esimerkiksi renderöinnistä). Taiteilija taas on vastuussa itse visuaalisen sisällön toteuttamisesta. Tämän takia on tärkeää, että taiteilija tulee toimeen tiimin teknisten osaajien kanssa. (Chandler 2014, 22.)

Taiteilijoita on erilaisia ja heitä erottaa heidän työtehtävänsä lisäksi heidän taidekoulutuksensa. Perinteisesti resurssitaiteilija (asset artist), animaattori ja tekninen taiteilija ovat olleet erillisiä ammattinimikkeitä, mutta nykyään kaikkien työt lasketaan teknisen taiteilijan töiksi. Täten suuremmissa projektiorganisaatioissa on yleensä kahdenlaista taiteilijoita: konseptitaiteilijoita ja teknisiä taiteilijoita. Konseptitaiteilija on klassisesti koulutettu, eli hän on oppinut piirtämään, maalaamaan ja muovaamaan taideteoksia kuten perinteinen taiteilija. Tekninen taiteilija on taas tekniseen taiteeseen erikoistunut taiteilija, jonka osaaminen koostuu 3D-mallinnuksesta, teksturoinnista ja animaatiosta. Pienemmissä organisaatioissa, etenkin jos on vain yksi taiteilija, taiteilija on usein tekninen taiteilija, mutta voi osata klassisiakin taitoja, mikä on suositeltavaa.

Etenkin projektiorganisaation johtavan taiteilijan tulisi osata sekä klassista että teknistä taidetta. Hän on johtavan insinöörin ja suunnittelijan kanssa osa ydintiimiä ja täten

vastuussa esituotannosta. Hän päättää johtavan suunnittelijan kanssa taidetyylistä, väripaaleista ja yleisesti ottaen pelin ulkonäöstä. Hän hallinnoi muita taiteilijoita ja valvoo taiteen laatua ja yhteenkuuluvuutta. Hän pitää huolen, että taide noudattaa suunnitelmaa. (Chandler 2014, 23.)

4.1.3 Suunnittelija

Suunnittelu on erittäin laaja vastuualue. Se kattaa peliprojektissa useamman tehtävän, kuten pelattavuuden, kontrolliskeeman ja hahmojen suunnittelun. Suunnittelijoiden tehtävänä on luoda pelaajille viihdyttävä pelikokemus. (Chandler 2014, 28.)

Koska suunnittelijoilla on niin suuri vastuu projektissa, isoissa projektiorganisaatioissa heitä on useampi. Suunnittelijoita voi olla yksi jokaisella vastuualueella. Vastuualueet ovat (Chandler 2014, 30):

- järjestelmäsuunnittelu, mikä kattaa esimerkiksi taistelujärjestelmän ja kontrolliskeeman
- käyttöliittymäsuunnittelu, mihin kuuluu enemmän kuin pelkästään graafisten käyttöliittymäelementtien suunnittelu
- kenttäsuunnittelu, missä taiteilijoiden ja insinöörien luomukset yhdistetään ja luodaan itse pelimaailma. Osaa koodata ja tehdä 3D-malleja.

Suunnittelijoiden joukkoon lasketaan lisäksi käsikirjoittaja (Chandler 2014, 30).

Käsikirjoitus voidaan joskus ulkoistaa, kun taas joskus sen tekee ihan tavalliset suunnittelijat. Yleensä käsikirjoittaja tai käsikirjoittajia hankitaan suuriin projekteihin, jos peli on erityisen tarinavetoinen. Käsikirjoittaja on taidoiltaan samanlainen kuin elokuvien käsikirjoittaja, mutta hänen tulee ottaa huomioon, että peli on vuorovaikutteinen kokemus. Pelien käsikirjoittaminen on hyvin erilaista työtä kuin elokuvien käsikirjoittaminen.

Joskus suunnittelija on entinen insinööri, joka on halunnut osallistua enemmän pelin suunnitteluun (Gregory 2009, 7). Suunnittelija voi kuitenkin olla koulutukseltaan mitä vain. On tunnettuja pelisuunnittelijoita, jotka ovat filosofian maistereita ja jotkut ovat puolestaan tulleet pöytäpelien puolelta peliteollisuuteen. Vaikka suunnittelijan työlle ei ole koulutusvaatimusta, se on luultavasti haastavin työ peliprojektissa.

Suunnittelija, etenkin johtava suunnittelija, on tiimistä se, joka on vastuussa projektin mittakaavasta eli scopesta. Hän on se, joka on vastuussa vision ja käytännön yhdistämisestä (Portnow 2008). Tämä vaatii projektihallinnollisia taitoja. Tämän lisäksi hänen tulee motivaation ylläpitämiseksi suunnitella peli siten, että tiimi pitää siitä (Portnow 2008). Jokaisen tiimin jäsenen tulee löytää jotain, mistä pitää pelissä (Schell 2015, 414).

Suunnittelijan tulisi myös ymmärtää pelaajaa, jotta pelistä saadaan yleisölle viihdyttävä peli (Adams 2010, 45). Hänellä pitää siis olla sekä viestintätaitoja että empatiaa.

Viihdyttävän pelikokemuksen aikaansaamiseksi voi tarvita kaikenlaista osaamista ja kokemusta. Ihan mikä vain kokemus tai tieto on hyödyllistä: historiasta antropologiaan, tekniikasta taiteeseen ja filosofiasta politiikkaan (Schell 2015, 3-4). Erityisen lahjakkaat suunnittelijat ovatkin usein kuin käveleviä ensyklopedioita. Tärkein taito on kuitenkin kuunteleminen: pitää osata kuunnella tiimiä, pelaajia ja itseään (Schell 2015, 5).

Suunnittelijan työ on haastavaa mutta olennaista peliprojekteissa. Vaikka kaikki osallistuvatkin suunnitteluun, suunnittelija on se henkilö, joka tietää mikä on viihdyttävää ja mikä ei. Hän ymmärtää mikä peleissä kookuttaa.

4.1.4 Testaaja

Testaajan rooli on vähiten osaamista vaativa (Chandler 2014, 33), mutta silti erittäin tärkeä. He eivät ole pelkästään vastuussa bugien löytämisestä pelin Alpha, Beeta ja julkaisukandidaatti –virstanpylväisiin, vaan he ovat myös vastuussa pelin laaduntarkistuksesta (Chandler 2014, 32). Yksi testaaja voi pelata projektin aikana peliä yli 1000 tuntia, joten heillä on yleensä paras käsitys siitä, onko peli viihdyttävä vai ei (Chandler 2014, 32).

Testaajia otetaan testaamaan pelin viihdyttävyyttä jo esituotannossa, mutta pääasiallisesti testaajien työ alkaa projektin tuotannossa (Chandler 2014, 32), jossa he testaavat vastaavako lopulliset virstanpylväät niihin suunniteltua sisältöä. Myöhemmin, tuotannon loppupuolella he ovat vastuussa pääasiallisesti bugien löytämisestä, jotta insinöörit voivat korjata ne. Tuotannon lopussa he ovat ainoita tiimin jäseniä jäljellä insinöörien lisäksi.

Testaajan alhaisen taitotasovaatimusten vuoksi tämä on yleisin tapa päästä pelialalle (Chandler 2014, 33). Tuottajat ovat esimerkiksi usein entisiä johtavia testaajia, jotka taas ovat entisiä testaajia. Heidän tulisi kuitenkin osata hyvin joitakin asioita kuten viestintää, koska testaaja tulee olemaan yhteydessä insinöörien kanssa läpi tuotannon (Chandler 2014, 33). Bugiraportteja tulee kirjoitettua useampi yhden projektin aikana.

Oleellinen ominaisuus testaajalle on kuitenkin se, että hänen tulisi olla pelaaja. Ollakseen testaaja, pitää olla usean vuoden kokemus pelien pelaamisesta. Testaajan kannattaa myös tietää yksityiskohtaisesti, mikä tekee pelistä viihdyttävän, kuten suunnittelija. Tähän auttaa peliteorian ymmärtäminen. Lisäksi johtavan testaajan tulisi osata henkilöstöhallintaa. (Chandler 2014, 33.)

4.1.5 Tuottaja

Tuottaja on peliteollisuuden nimitys projektipäällikölle. Hänen tehtävänä on pitää huolta siitä, että tiimillä ei ole esteitä kehitykselle. Hän hoitaa aikataulut, neuvottelut, kokoukset, tapaamiset rahoittajan kanssa ja demotilaisuudet (Chandler 2014, 18). Hän on välikätenä kehitystiimin ja pelitalon johdon välillä (Chandler 2014, 18). Hän on tiimin pääsuunnittelijan lisäksi pelin visiosta kaikista tietoisin projektiryhmän jäsen. Hänen tehtävänä on saada markkinoille mahdollisimman hyvä peli.

Tuottajan rooli on haastava samalla tavalla kuin projektipäällikön. Hänellä pitää olla erinomaiset johtajuus-, viestintä- ja projektinhallintataidot (Chandler 2014, 21). Hänen pitäisi pystyä ratkomaan konfliktitilanteet ja motivoimaan tiimiä (Chandler 2014, 21). Tämän lisäksi on suositeltavaa, että hän osaa taiteen ja koodaamisen perusteet. Tuottaja on mahdollisesti projektiryhmän yksi taidokkaimmista jäsenistä.

Pelitaloissa tuottaja on usein peliprojektien konkari, joka on ollut pelialalla useamman vuoden ja on ajan myötä siirtynyt projektinhallinta tehtäviin. Pelitaloissa esimiehenä on usein tuottaja. Tuottaja työskentelee myös usein useamman pelin parissa samaan aikaan.

4.1.6 Äänituotanto

Äänituotannossa työskentelee pelin musiikin säveltäjä sekä äänisuunnittelija (Chandler 2014, 31). Äänisuunnittelija on äänitehosteisiin erikoistunut työntekijä: hän päättää pelin äänimaailmasta suunnittelijoiden kanssa ja mahdollisesti tuottaa äänitehosteet itse, omassa studiossaan (Chandler 2014, 31). Säveltäjä taas on sama työnkuvaltaan kuin säveltäjä elokuvateollisuudessa, joskin peleissä musiikin pitää esimerkiksi pystyä toistumaan saumattomasti.

Äänituotanto usein ulkoistetaan kolmannelle osapuolelle, paitsi silloin kun on kyseessä suurempi pelitalo. Suuremmalla pelitalolla, joka tuottaa suuremman budjetin pelejä, on usein oma äänituotantotiimi sekä studio.

4.2 Tiimikokoonpanot

Kehitystiimin kokoonpano riippuu tarjolla olevasta henkilöstöstä, pelin sisällön mittakaavasta ja projektin budjetista. Suurissa projektiorganisaatioissa voi jokaista pelin ominaisuuksien osa-aluetta kohtaan on yksi edustaja jokaisella osastolla. Tarkoituksena on, että jokaista ominaisuutta kohtaan olisi asiantuntija, joka on erikoistunut kyseiseen alaan, olisi se sitten tekoäly tai pelattavuus.

On tärkeää muistaa, että taideresurssien luomiseen menee huomattavasti enemmän aikaa kuin koodaamiseen. Insinööri saa siis oman tehtävänsä aikaisemmin valmiiksi kuin taiteilija. Tiimin koostamisessa ja sen aikataulutuksessa tämä kannattaa ottaa huomioon hankkimalla vaikkapa kaksinkertainen määrä taiteilijoita insinööreihin nähden, eli kaksi taiteilijaa yhtä insinööriä kohtaan. Täten hyvä tiimin roolien jakautuminen olisi 50% taitelijoita, 25% insinöörejä ja 25% suunnittelijoita.

Pienessä tiimissä tai vähäisen hierarkian organisaatioissa yksi tiimin jäsen voi suorittaa useamman osaston töitä. Jos esimerkiksi insinööri osaa tehdä 3D-malleja, hän voi tehdä molempia sen mukaan, miten aikataulu sallii. Tällainen projektiorganisaatio on tuttu ketteristä kehitysmenetelmistä perinteisestä ohjelmistokehityksessä. Tämä kuitenkin vaatii enemmän moniosaajuutta.

Tärkeintä on, ettei yksi tiimin jäsen ole yksin vastuussa suuremmasta määrästä töitä kuin muut. Pelin tuotantolinjan (asset pipeline) suorituskyvyn parantamiseksi kannattaa pyrkiä järjestämään henkilöstö siten, että tyhjäkäyntiä saadaan vähennettyä. Kannattaa myös hankkia tarvittavat asiantuntijat sen mukaan, mitä ollaan tekemässä. Jos voidaan valita kahden taiteilijan väliltä, tulisi valita hänet, joka osaa paremmin pelin visuaalisen tyylin ja sopii tiimiin paremmin persoonallisuudeltaan.

5 Ketterät kehitysmenetelmät

Viime aikoina on yleistynyt ketterien kehitysmenetelmien käyttö peliprojekteissa, etenkin sitä mukaan kun uusia työntekijöitä saapuu peliteollisuuteen perinteisiltä ohjelmistoteollisuuden aloilta. Etenkin Scrum -ketteräkehityskehikko on yleistynyt huomattavasti.

Scrumissa kehitetään ominaisuuksia iteratiivisesti sykleissä, jota kutsutaan sprinteiksi. Yksi sprintti voi kestää viikosta neljään viikkoa, mutta yleensä noin kaksi viikkoa. Yhdessä sprintissä kehitetään kokoelma käyttötapauksia, jotka on valittu sprintin suunnittelukokouksessa. Yhdessä sprintissä käyttötapaus määritellään, toteutetaan ja testataan. Käyttötapaukset valitaan ylläpidettävästä backlogista, missä on kaikki toteutetut ja toteutettavat käyttötapaukset. Ominaista Scrumille on ominaisuuksien nopea iteroiminen.

Peliprojektien tekninen osuus soveltuu hyvin Scrumiin, joka on peräisin perinteisemmiltä ohjelmistokehityksen aloilta, sen sijaan taide ei sovellu siihen yhtä hyvin pitkän keston takia. Tämä vaatii ominaisuuksien erilaista mitoittamista sprinteille: yhdellä sprintillä ei

välttämättä saa toteutettua yhtä monta käyttötapausta kuin esimerkiksi web-aplikaation kehityksessä. Taide on oma ominaisuutensa myös, ja vaatii useamman iteraation. Scrumin soveltaminen peliprojektissa vaan vaatii uudenlaista ajattelua ominaisuuksien osalta.

Muiden ketterien kehitysmenetelmien, kuten Double Diamondin ja Kanbanin, soveltaminen vaatii samanlaista ajattelun muuttamista. Menetelmän soveltamista pitää ajatella taiteen näkökulmasta, mihin menee aikaa ja vaatii useamman iteraation.

6 Rogue Runner -projektin elinkaari

Rogue Runner on mobiilipeli joka on kehitetty Unity -pelikehitysokalulla, mikä on tällä hetkellä jälkituotannossa. Esituotanto ja tuotanto suoritettiin Softala projekti III -kurssilla keväällä 2015. Pelin kehitti Otto Uusihakala, Lauri Saarelainen, Sami Soikkeli ja Laura Pimiä.

Rogue Runner on genreltään 2D tasoloikan alalaji, nimeltään endless runner. Ikuisesti jatkuva kenttä saavutetaan satunnaisluonnilla: viidestä mahdollisesta osasta valitaan satunnaisesti yksi, joka laitetaan kentän perään. Mitä pidempään pelaaja onnistuu pitämään pelaajahahmon juoksemassa, eikä tipu kuiluihin tai osu kiviin, sitä enemmän pisteitä hän saa. Lisäpisteitä saa pistekääröistä, joita löytää matkan varrelta.

Seuraavaksi tutkitaan miten projekti eteni ja miten onnistuttiin.

6.1 Esituotanto

Tiimillä oli jo konsepti pelistä ennen kuin projektin hyväksyntää oli haettu. Kun projekti hyväksyttiin, siirryttiin pikaisesti muun esituotannon pariin. Esituotannossa vietettiin noin kaksi viikkoa. Sen aikana saatiin osittain valmiiksi pelisuunnitelma (GDD, Game Design Document).

Tärkeää oli todistaa, että tiimi pystyi projektiin, koska tuloksia oltiin luvattu jo ensimmäisenä päivänä kun projekti hyväksyttiin. Tärkeää oli saada jotain näkyvää aikaiseksi. Jo hyvin varhaisessa ajassa kehitettiin digitaalinen prototyyppi, missä hahmo liikkui. Prototyypin taideresurssit olivat tarkoituksella hyvin puutteellisia, koska ne olivat väliaikaisia.

Esituotanto huipentui kokoukseen kurssin opettajien kanssa, jossa esiteltiin lopullinen prototyyppi. Prototyyppiin mennessä oltiin saatu aikaiseksi pelin keskeiset ominaisuudet: pelaajahahmon liikkuminen, hyppääminen ja kentän satunnaisluonti. Esituotannon tuloksiin oltiin hyvin tyytyväisiä ja projektissa siirryttiin esituotantoon.

6.2 Tuotanto

Tuotannossa tärkeintä oli parantaa pelin laatua. Vaikka pelin keskeiset ominaisuudet olivatkin valmiita, niiden laatu oli keskinkertaista. Lisäksi puuttui joitakin ominaisuuksia, kuten pistetaulukko ja asetukset. Taidevastaava Laura aloitti pelaajahahmon animaation työstämisen sekä konseptitaiteeseen pohjautuvan kentän taideresurssien työstämisen.

Hahmon hypyn tuntumaa parantaakseen hahmon ohjauslogiikkaa muutettiin hyödyntämään pelin fysiikkaa eri tavalla ja painovoiman vaikutusta lisättiin. Tavoitteena oli hypyn kaaren muokkaaminen loivasta pitkästä hypystä lyhyeksi korkeaksi hypyksi. Tavoite saavutettiin, joskin kentät piti suunnitella uusiksi hypyn pituuden ja korkeuden takia.

Tuotannon loppupuolella testauksessa paljastui pelin keuhno suorituskyky. Pelin optimoinnin puute oli johtanut siihen, että peli ei toiminut lainkaan edes vuoden vanhoilla älypuhelimilla, vaikka kohdealustana oli Android-älypuhelimet. Optimoinnin tarvetta ei missään vaiheessa huomioitu ja testausta oli laiminlyöty. Ongelmat löydettiin niin myöhään tuotannossa, ettei niitä saatu korjattua kurssin loppuun mennessä.

6.3 Jatkokehityssuunnitelmat

Kesällä ja varhaisryksyillä tiimi jatkoi suunnittelua pelin jatkokehityksen osalta. Päätettiin, että projektia jatkettaisiin ainakin siihen asti, että saataisiin peli pelattavaan kuntoon ja Google Play -levitysalustalle. Ajankohtaa ei päätetty, mutta tiimi suunnitteli pyrkivänsä saamaan uuden version aikaiseksi ennen joulua. 28.11.2015 valmistui uudempi versio pelistä, missä oli uudelleen toteutettuna kentän satunnaisuontti. Peli myös hyödyntää vähemmän olioita, mikä vähentää muistikäyttöä huomattavasti uuden satunnaisuonttilogiikan kanssa.

Testaus kuitenkin lykkääntyy 2016 ensimmäiselle kvartaalille (Q1). Silloin tiimi testaa uusiksi suunniteltuja kenttiä ja pelin suorituskykyä pelitestaajilla, jotta saataisiin jälleen uutta tietoa pelin kehittämiseksi.

6.4 Mikä meni hyvin? Mikä meni huonosti?

Tiimi piti projektia alusta lähtien kouluprojektina, joten tiimi ei ole missään vaiheessa miettinyt kohdeyleisöä. Tiimillä oli myös liiallinen kiire päästä pois esituotannosta, mikä johti pelisuunnitelman puutteellisuuteen. Pelisuunnitelmaa ei päivitetty tuotannossa muuta kuin alkupuolella.

Myös puuttellinen ymmärrys tekniikan osalta johti optimoinnin puutteen lisäksi vaikeasti luettavaan koodiin. Yksi muutos koodiin pystyi rikkomaan koko pelin. Mitään ohjelmointikäytäntöjä ei asetettu. Yksi henkilö teki suurimman osan pelin logiikasta (pääkoodaaja), joten muiden ymmärrys pelin logiikasta jäi puutteelliseksi.

Pelisuunnittelua ei koskaan otettu vakavasti, joten ensimmäisten pelitestauskierroksien palaute pelin viihdearvosta oli hyvin kielteistä. Vaikka moni pitikin pelistä ja kertoi sen olevan koukuttava, tuli hyvin selväksi, että peli on liian vaikea. Useasti haaste pelissä johtui siitä, että tapahtui jotain odottamatonta. Pelaaja esimerkiksi ei tiennyt ensimmäisellä kerralla, että tulossa on joukko esteitä, mitkä pitää väistää tietyllä tavalla.

Projektissa onnistuttiin saamaan hyvin aikaisin lupaavia tuloksia, mitkä todistivat tiimin kuitenkin osaavan riittävästi projektin suorittamiseksi. Vaikka esituotannon tuotokset olivatkin puutteellisia, prototyyppi oli erittäin lupaava jopa teknisesti eikä esituotannossa vietetty liikaa aikaa. Projekti pysyi aikataulussa ja luvatut virstanpylväät saatiin aikaiseksi määräaikoihin mennessä. Viestintä tiimin jäsenten välillä sujui ongelmitta ja yhteishenki oli vahva. Motivaatio pysyi hyvänä projektin läpi ja kaikki pyrkivät yhdessä saamaan virstanpylväät valmiiksi.

7 Johtopäätökset ja pohdinta

Pelien kehittäminen on enemmän ohjelmistokehitystä kuin elokuvien tuottamista, mutta se silti on erilaista kuin perinteinen ohjelmistokehitys. Viihdearvon tärkeys pelien menestymisessä tekee siitä erilaisen ohjelmistomuodon kuin muut vastaavat. Kun muut ohjelmistot pyrkivät pääasiassa ratkaisemaan käytännön ongelman, pelit ovat viihdettä ja niiden ratkaisema ongelma on viihteen puute.

Pelituotteen kehityksen elinkaari on näennäisesti samanlainen kuin muilla ohjelmistokehityksen aloilla, mutta sen erottaa se, että siinä otetaan huomioon kaksi muuta alaa ohjelmoinnin lisäksi. Nämä ovat taide ja suunnittelu, jotka tuovat peleihin niiden luovuuden ja viihteen. Pelien kehittämisessä ohjelmointi on tämän luovuuden kääntämistä pelilogiikaksi.

Vaikka ohjelmoinnin päämäärä pelien kehittämisessä onkin eri, peliohjelmointi on haastava ala ja vaatii paljon perinteisten algoritmien ja suunnittelumallien ymmärtämistä ja soveltamista. Esimerkiksi yleisin peleissä käytetty reitinhakualgoritmi, A*-reitinhakualgoritmi, on peräisin vuodelta 1968, mikä on laajennus Edsger Dijkstran vuonna 1959 luomaan reitinhakualgoritmiin. Web-sovelluskehityksessä näitä perinteisiä suunnittelumalleja ja algoritmeja harvemmin tarvitsee, mutta pelejä kehittäessä esimerkiksi oliollas on peleissä pakollinen suorituskyvyn ylläpitämiseksi. Suorituskyky onkin paljon suuremmassa osassa peleissä kuin web-sovelluskehityksessä.

Yleinen väärinkäsitys on, että pelien suunnittelu on helppoa. Usein kuitenkin paljastuu, että vaikka onnistuisi suunnittelemaan jotakin, joka viihdyttää itseänsä, ei välttämättä viihdytä muita. Tämän takia pelialan asiantuntijat käyttävätkin pelien suunnittelussa pelaaja-keskeistä suunnittelua, jossa peli suunnitellaan hypoteettiselle kohdeyleisön edustajalle. Kohdeyleisö on siis paljon suuremmassa osassa pelien kehitystä kuin luulisi. Jos peli ei viihdytä kohdeyleisöä, peli on epäonnistunut.

Taide on aihe, jossa kaikki ovat kuitenkin samaa mieltä. Mitä parempaa taide on, sitä parempi peli on. Taiteilijan tai taiteilijoiden taidot ovatkin hyvin olennaisia pelien kehityksessä. Etenkin, kun peli ensimmäisen kerran esitellään maailmalle vaikkapa kuvakaappauksilla ja trailerilla, pelin ulkonäkö on ensimmäinen asia, mistä mahdollisella asiakkaalla on ensimmäisenä mielipide. Kehittäjällä on yksi mahdollisuus vakuuttaa, ja onnistumisen todennäköisyys on huomattavasti parempi kun taide on hyvää.

Taiteen luomiseen kuitenkin menee huomattavasti aikaa. Insinöörit, joille muut ohjelmistokehityksen alat ovat tuttuja, voivatkin joutua hankaluuksiin suunnitellessaan

aikataulua, joka ottaa huomioon taiteeseen menevän ajan. Aina tule ajatelleeksi, kuinka paljon aikaa menee laadukkaan taiteen luomiseen. Onkin suositeltavaa, että muulta alalta pelialalle saapuva ohjelmistokehittäjä tutustuu taiteeseen ja sen tuottamiseen. Tämä auttaa myös huomattavasti, kun pitää olla yhteydessä taiteilijoiden kanssa.

Jotta suurin osa tekniseen ja taiteelliseen toteutukseen liittyvät ongelmat voidaan välttää, on tärkeää pysyä riittävän pitkään esituotannossa ja löytää vastaukset kaikkiin kysymyksiin. On parempi olla varma siitä, mitä on tekemässä, kuin olla puoliksi tietoinen siitä, mitä on tekemässä. Huonosti suoritettu esituotanto on yleisin ongelmien lähde. Jopa huono tiimin jäsenen valinta johtuu virheestä esituotannossa: henkilöstösuunnitelma hoidetaan esituotannossa, joten tuotannossa oleva henkilöstö on päätetty jo esituotannossa. Yleinen virhe on jättää testaukselle liian vähän aikaa tai olla suorittamatta sitä riittävästi vaikka aikaa olisikin. Miten kukaan voi tietää onko peli esituotannon suunnitelman mukainen jos peliä ei testata ahkerasti? Tuloksena voi olla erittäin huono peli, joka ei viihdytä mahdollisia asiakkaita.

Pelien kehityksessä on olennaista ymmärtää paljon asioita, teknisiä ja taiteellisia, mutta tärkeintä on ymmärtää, että pelit ovat viihdettä. Pelin viihdearvo riippuu pelin elinkaaren oikeanlaisesta suorittamisesta. Nykyään kuka tahansa voi tehdä pelejä, mutta ainoastaan intohimoiset tiimit, jotka ymmärtävät pelin elinkaaren vaiheiden tehtävät ja niiden tärkeyden, tekevät hyviä pelejä.

Jos tehdään virheitä, kehittäjän tulisi oppia virheestään ja suorittaa seuraava projekti paremmin. Jos ongelma oli esimerkiksi ettei esituotannossa pysytty riittävän pitkään, on syytä pysyä seuraavalla kerralla esituotannossa niin pitkään, että ollaan aivan varmoja että se on valmis. Jos testaus epäonnistuu, asia otetaan huomioon seuraavan projektin esituotannossa ja testaussuunnitelma tehdään paremmin.

Lähteet

Adams, E. 2010. Fundamentals of Game Design. 2. painos. New Riders. Berkeley CA.

Autodesk. 2015. Autodesk Navigation. Luettavissa:

<http://gameware.autodesk.com/navigation/features>. Luettu: 25.10.2015.

Chandler, H.M. 2014. Game Production Handbook. 3. painos. Jones & Bartlett Learning. Burlington MA.

Gregory, J. 2009. Game Engine Architecture. 1. painos. CRC Press. Broken Sound Parkway NW.

Portnow, J. 2008. Scope: A Lesson in Game Design. Luettavissa:

http://www.gamecareerguide.com/features/508/scope_a_lesson_in_game_design.php.

Luettu: 2.9.2015.

Schell, J. 2015. The Art of Game Design – A Book of Lenses. 2. painos. CRC Press. Broken Sound Parkway NW.

Sony Computer Entertainment Inc. 2011. PLAYSTATION2 sales reach 150 million units worldwide. Luettavissa: http://www.scei.co.jp/corporate/release/110214_e.html. Luettu: 2.12.2015.

Sony Computer Entertainment Inc. 25.11.2015. PlayStation®4 (PS4™) Sales Surpass 30.2 Million Units Worldwide. Lehdistötilaisuus. Tokio, Japani.

Stevens, M. 2013. A brief history of video game consoles from the early 1970s to the late 1990s. Luettavissa: <http://venturebeat.com/community/2013/12/30/a-brief-history-of-video-game-consoles-from-the-early-1970s-to-the-late-1990s/>. Luettu: 2.12.2015.

Vankka, E. 2014. Free-To-Play Games: Professionals' Perceptions. University of Tampere. Master's Thesis. Luettavissa:

<http://tampub.uta.fi/bitstream/handle/10024/95105/GRADU-1395760771.pdf?sequence=1>.

Luettu: 4.11.2015.