

Sami Nissinen

Mobiilipelin kehitys Androidille Unityllä

Mobiilipelin kehitys Androidille Unityllä

Sami Nissinen
Opinnäytetyö
Syksy 2021
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinto-ohjelma, Web-sovelluskehitys

Tekijä: Sami Nissinen
Opinnäytetyön nimi: Mobiilipelin kehitys Androidille Unityllä
Työn ohjaaja: Teppo Räisänen
Työn valmistumislukukausi ja -vuosi: Syksy 2021
Sivumäärä: 24

Opinnäytetyön tavoitteena oli tehdä hyperkasuaali mobiilipeli Unityllä. Pelissä ohjataan järkälettä loppumattomalla pelikentällä väistellen esteitä ja keräten esineitä. Toimeksiantajana oli Oulu Game Labilla syntynyt Lost Hoodie Oy. Toimeksiantaja keskeytti toimintansa opinnäytetyön tekemisen aikana, mikä vaikeutti työn tekemistä ja johti siihen, ettei kaikkia suunniteltuja toimintoja saatu lisättyä peliin.

Opinnäytetyössä käydään läpi mobiilipelien historiaa ja matkapuhelinta sovellusalustana peleille teknologian ja matkapuhelinmarkkinoiden kehitystä seuraten. Mobiilipelaamisen nykytilaa käydään myös lyhyesti läpi. Tämän lisäksi opinnäytetyössä kerrotaan hieman Androidin historiasta, arkkitehtuurista ja Google Playstä, joka on Googlen virallinen sovelluskauppa Android-laitteilla. Tämän jälkeen opinnäytetyössä kerrotaan Unityn käyttämisestä, jonka jälkeen käydään läpi itse pelin toteutus.

Mobiilipelien historiasta löytyi runsaasti tietoa eri verkkosivustoilta ja peleihin keskittyvistä lehdistä. Etenkin PocketGamer.Biz-sivuston A Brief History of Mobile Games -artikkelisarja oli erittäin hyödyllinen. Androidista löytyi myöskin runsaasti tietoa Androidin viralliselta sivustolta ja eri verkkojulkaisuista. Unityn omat verkkomanuaalit ja tutoriaalit olivat korvaamaton apu Unity-osiossa ja itse peliä tehdessä. Pelisuunnittelussa apuna oli Tony Mannisen Ideasta eteenpäin -kirja. Suurin apu oli kuitenkin Lost Hoodien työntekijöiden aiemmista peleistään kartuttama tietotaito ja kokemus.

Pelistä saatiin tehtyä toimiva versio, jossa olivat perusominaisuudet kuten järkäleen ohjaus sivusuunnassa ja kiihdytys, törmäysten tunnistus ja ilmoitus pelin päättymisestä sekä pisteiden laskeminen ja huippupistemäärän ilmoitus. Kenttäeditori ja monipuolisemmat esteet jäivät pois, koska ne olivat liian työläitä toteuttaa käytettävissä olevassa ajassa. Peliä ei voitu lisätä Google Play -sovelluskauppaan, koska Lost Hoodie keskeytti toimintansa opinnäytetyön tekemisen aikana. Mainosten näyttämistä ja analytiikkatietojen keräämistä ei voitu toteuttaa samasta syystä.

Opinnäytetyön tekeminen antoi hyvät pohjatiedot, kuinka Unityllä voidaan tehdä mobiilipelejä. Opittuja tietoja voidaan hyvin käyttää jatkossa uusien mobiilipelien tekemiseen tai opinnäytetyön aikana tehdyn pelin jatkokehittämiseen, jos Lost Hoodie Oy vielä jatkaa toimintaansa.

Asiasanat: Android, Unity, mobiilipeli

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author: Sami Nissinen
Title of thesis: Mobile game development for Android in Unity
Supervisor: Teppo Räisänen
Term and year when the thesis was submitted: Autumn 2021
Number of pages: 24

The aim of the thesis was to make a hyper casual mobile game for Android with Unity. The player controls a boulder on an endless level by dodging obstacles and collecting objects. The client was Lost Hoodie Oy, formed at Oulu Game Lab. The client suspended its operations during the making of the thesis, which hindered the process and as a result not all the planned features could be added to the game.

The thesis also reviews the history of mobile games, and the mobile phone as a platform for games following the development of technology and the mobile phone market. The current state of mobile gaming is also mentioned. In addition to this, the thesis briefly tells the history of Android, its architecture, and Google Play, which is Google's official app store for Android devices, as well as a place to buy digital media. Next, the thesis briefly explains the use of Unity, after which we go through the implementation of the game itself.

There is a wealth of information on the history of mobile games on various websites and in game-focused magazines. The Brief History of Mobile games series on PocketGamer.Biz was very useful. A lot of information about Android was found on the official website of Android and in various websites such as VentureBeat. Unity's own online manuals and tutorials were an invaluable aid in the Unity section and in making the game itself. Tony Manninen's book *Ideasta eteenpäin* was an aid in designing the game. The biggest help, however, was the know-how and the experience the Lost Hood employees had gained from their previous games.

A functional version of the game was made, with basic features such as lateral control and acceleration of the boulder, collision detection and end-of-game notification, as well as scoring and high score reporting. Level editor and more varied obstacles were omitted because they were too laborious to implement in the time available. The game could not be added to the Google Play App Store because Lost Hoodie's operations were suspended while completing the thesis. It was not possible to display ads and collect analytics data for the same reason.

Completing the thesis provided a good basic knowledge of how Unity can be used to make mobile games. The learned information can well be used in the future to make new mobile games or to further develop the game made during the thesis, if Lost Hoodie Oy is continuing its operations.

Keywords: Android, Unity, Mobile game

SISÄLLYS

1	Johdanto	6
2	Mobiilipelit	7
2.1	Historia	7
2.2	Nykypäivä	9
3	Android	11
3.1	Historia	11
3.2	Arkkitehtuuri	11
3.3	Google Play	13
4	Unity	14
4.1	Käyttöliittymä	14
4.2	Unityn käyttäminen	15
5	Pelin toteutus	17
5.1	Pelin idea	17
5.2	Pelin toteutus	17
5.3	Pelin graafinen ilme	19
5.4	LevelSpawner-toiminnallisuus	20
6	Pohdinta	22
	LÄHTEET	23

1 JOHDANTO

Aihe tähän opinnäytetyöhön syntyi alun perin Oulu Game Labissa syntyneen Lost Hoodie Oy:n toimeksiannosta. Tämän opinnäytetyön tarkoituksena on kehittää hyperkasuaali mobiilipeli Android-alustalle. Hyperkasuaalit pelit ovat hyvin minimaalisia mobiilipelejä, jotka voidaan yleensä aloittaa nopeasti lataamisen jälkeen ilman tutoriaalien katsomista tai ohjeiden lukemista. Ne ovat yleensä ilmaisia ja niiden tuotto tulee pääsääntöisesti niissä näytettävistä mainoksista ja pelin sisäisistä ostoksista.

Mobiilipelimarkkinoiden arvo on kasvanut joka vuosi älypuhelinien julkaisun jälkeen. Tänä päivänä mobiilipelaaminen on markkinoiltaan suurempi kuin konsoli- ja PC-pelaaminen yhdessä. (Wallach 2020, viitattu 13.8.2021.)

Tänä päivänä pelikehityksessä käytetään yleensä valmiita pelikehitystyökaluja. Muutamia esimerkkejä näistä ovat mm. Unity, Unreal ja Game Maker Studio. Valmiin pelimoottorin ja kehitysympäristön käyttäminen nopeuttaa pelikehitystä, koska resursseja ei tarvitse käyttää oman pelimoottorin suunnittelemiseen ja ohjelmoimiseen. Pelikehitystiimit koostuvat yleensä eri alojen asiantuntijoista, jotka vastaavat pelin eri osa-alueista. Kehitystiimin rooleja ovat mm. tuottaja, pelisuunnittelija, konseptikuvittaja, art director, äänisuunnittelija, kenttäsuunnittelija, animaattori ja ohjelmoija. Pienissä tiimeissä kullakin jäsenellä on yleensä useita rooleja. (Manninen 2007, 46–50.)

Opinnäytetyön luvussa 2 kerrotaan mobiilipeleistä, niiden historiasta ja nykytilasta. Tämän jälkeen luvussa 3 esitellään tämän päivän suosituin mobiilikäyttöjärjestelmä Android ja käydään läpi sen historiaa ja arkkitehtuuria. Luvussa 4 esitellään tässä opinnäytetyössä käytettyä Unity-pelimoottoria. Seuraavaksi luvussa 5 käydään läpi tämän opinnäytetyön aiheena olleen pelin tekeminen ideointi- ja suunnitteluvaiheesta käytännön toteutukseen. Lopuksi luvussa 6 pohditaan mitä lopulta saatiin aikaan, mitä haasteita kehitystyössä oli ja mitä tästä kaikesta opittiin.

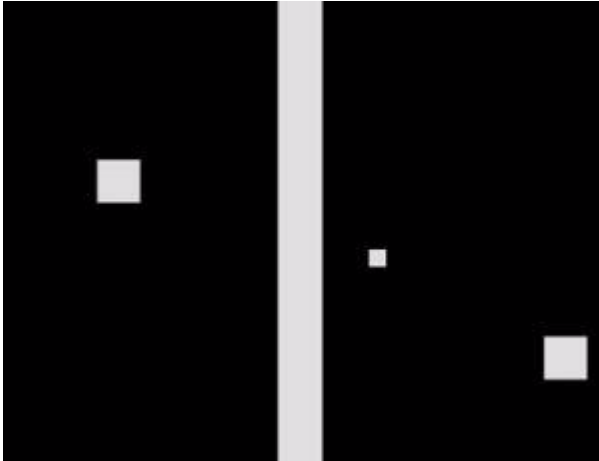
2 MOBIILIPELIT

Kappaleessa 2.1 käydään läpi mobiilipelien historiaa ja matkapuhelimia sovellusalustana peileille teknologian ja matkapuhelinmarkkinoiden kehitystä seuraten. Kappaleessa 2.2 käydään läpi lyhyesti mobiilipelaamisen nykytilaa ja sen erityispiirteitä etenkin konsolipelaamiseen verrattuna ja mobiilipelimarkkinoita.

2.1 Historia

Koko ihmiskunnan historian ajan peleillä on ollut tärkeä osa kaikissa kulttuureissa. Tiettyjen eläinten luiden suurempaa esiintyvyyttä muihin luihin verrattuna arkeologisissa löydöksissä on pidetty merkinä siitä, että niitä on käytetty pelinappuloina joissain esihistoriallisissa peleissä (Koerper & Whitney-Desautels 1999, 70). Ihmiskunnan ymmärryksen ympäröivästä maailmasta kehittytyä monimutkaisemmaksi myös ihmisten kehittämät työkalut ja sitä myötä pelit kehittyivät monimutkaisemmiksi.

Ensimmäinen videopeli syntyi 1958 lokakuussa, kun Brookhaven National Laboratoryn johdossa tuolloin ollut fyysikko William Higinbotham sai ajatuksen kehittää pelin avoimien ovien päville herättämään yleisön kiinnostusta laboratorion toimintaan. Peli oli nimeltään Tennis for Two ja sitä pelattiin oskilloskoopin ruudulla valitsemalla säätimellä mailan kulma ja painamalla nappia oikeaan aikaan palloa lyödäkseen. (APS News 2008, viitattu 24.10.2020.) Ensimmäinen kaupallinen videopelikonsoli on Magnavox Odyssey, joka julkaistiin 1972 (Baer 2005, 25). Konsoli liitettiin televisioon ja se pystyi näyttämään ruudulla pystysuoran viivan ja kolme neliön muotoista kappaletta, joista pelaajat pystyivät ohjaamaan kahta ja konsoli yhtä. (Kts. kuva 1.) Pelikonsoliin pystyi liittämään game cardeiksi kutsuttuja piirilevyjä, jotka muuttivat ruudulla näkyvien osien toimintaa sekä niiden reagoitua ohjainten syötteisiin mahdollistaen eri pelien pelaamisen yhdellä konsolilla. Yksi Magnavoxin mukana tullut peli oli nimeltään Ping Pong, johon Atari perusti kuuluisaksi tulleen Pong-kolikkopelin.



Kuva 1. Magnavox Odysseyn Ping Pong vuodelta 1972.

Videopelit kehittyivät nopeasti samalla, kun niitä pyörittävä laitteisto kehittyi. Matkapuhelinten kehittymisen myötä myös niihin alettiin kehittää pelejä. Ensimmäinen suuren yleisön tietoisuuteen päätynyt mobiilipeli oli Nokian vuonna 1997 julkaistussa 6110:ssa ja useimmissa Nokian sen jälkeen julkaistuissa puhelimissa ollut Snake (Wright 2016a, viitattu 16.8.2021).



Kuva 2. Nokian Snake-peli.

Kuvassa 2 näkyy Snake-peli Nokian 3110 kännykän ruudulla. Pelissä ohjataan käärmettä, joka yrittää syödä ruudulle ilmestyviä nappuloita samalla koettaen väistellä ruudun seiniä ja omaa var-taloaan, joka pitenee jokaisen nappulan syömisen jälkeen.

Snake ja muut sen aikaiset mobiilipelit olivat kovakoodattuja matkapuhelinten käyttöjärjestelmään. WAP (Wireless Application Protocol) on käytöstä poistunut langattomien sovellusten protokolla, joka mahdollisti yksinkertaisten Internet-sivujen näyttämisen pienellä matkapuhelimiin kovakooda-tulla selaimella. Tätä teknologiaa alettiin hyödyntää myös mobiilipeleissä. Vuonna 1999 Nokia

julkisti ensimmäisen WAP-selaimellisen matkapuhelimen Nokia 7110. Koska pelit pyörivät Internetissä palvelimilla, joihin otettiin yhteyttä matkapuhelimen WAP-selaimella, ne eivät voineet sisältää liikkuvaa grafiikkaa tai nopeatempoista toimintaa. Verkkoyhteys mahdollisti kuitenkin vuoropohjaisen moninpelien pelaamisen. (Wright 2016a, viitattu 16.8.2021.)

WAP-pelit jäivät lyhyehköksi välivaiheeksi ennen kuin J2ME (Java 2 Micro Edition) -ympäristöstä tuli vallitseva pelialusta mobiililaitteilla Euroopassa ja etenkin Suomessa. Ensimmäiset Javaa tukevat matkapuhelimet Euroopassa olivat Nokia 3410 ja Siemens M50. Käyttäjät pystyivät itse asentamaan Java-pelejä J2ME:tä tukeviin puhelimiin. (Wright 2016b, viitattu 17.8.2021.) J2ME-ohjelma suoritetaan puhelimissa Java-virtuaalikoneella, joka toimii ohjelmallisesti laitteiston ja käyttöjärjestelmän päällä. Tämän takia ne toimivat hitaammin kuin käyttöjärjestelmän natiivit ohjelmat.

2000-luvun alkupuolella tulivat älypuhelimet, joihin pystyi asentamaan pelejä ja muita ohjelmia, jotka ajetaan natiivisti samaan tapaan kuin puhelimen omat esiasennetut ohjelmat. Älypuhelin alkuajankana markkinoita hallitsivat Blackberry, Symbian ja Windows Mobile -käyttöjärjestelmät. Puhelinten näyttöjen koot, tarkkuudet ja värimäärät kehittyivät puhelimen muiden ominaisuuksien mukana ja pelit luonnollisesti käyttivät hyväkseen parantuneita ominaisuuksia. Alkuun suurimmassa osassa älypuhelimia oli edelleen näppäimistö, vaikka joitakin kosketusnäyttöön perustuvia puhelimia nähtiinkin. Siirtyminen kosketusnäyttöä käyttäviin puhelimiin kiihtyi, kun Applen iPhone julkaistiin vuonna 2007 (Vogelstein 2008, viitattu 22.8.2021). Seuraavana vuonna markkinoille tulivat ensimmäiset Android-puhelimet. Applen iPhone-puhelimet ja Googlen Android-käyttöjärjestelmää käyttävät puhelimet hallitsevat nykyään lähes täydellisesti matkapuhelinmarkkinoita. Vuonna 2008 Applen julkistama sovelluskauppa App Store mullisti mobiilisovellusten ja -pelien jakelun tarjoamalla keskitetyn paikan löytää pelejä ja muita sovelluksia suoraan matkapuhelimesta (Bonnington 2013, viitattu 25.8.2021). Google julkisti oman sovelluskauppansa pian sen jälkeen.

2.2 Nykypäivä

Matkapuhelimet ovat kehittyneet ominaisuuksiltaan niin paljon, että konsoli- ja mobiilipelaamisen ero on hämärtynt. Jopa keskihintaluokan puhelimet vastaavat suorituskyvyiltään ja grafiikkaominaisuuksiltaan muutamaa sukupolvea aikaisempaa pelikonsolia.

Konsolipelaamisesta poiketen mobiilipelaamiselle tunnusomaista on kosketusnäytön käyttäminen pelin kontrolloimiseen eikä erillistä peliohjainta tarvita, vaikka monet pelit niitä tukevatkin. Mobiilipeleissä voidaan käyttää myös erikoisempia kontrollointitapoja, jotka perustuvat matkapuhelimissa oleviin sensoreihin. Lokaatiopohjaisissa peleissä peliä ohjataan liikkumalla fyysisesti oikeassa maailmassa. Suosituin lokaatiopohjainen mobiilipeli on Pokémon GO (Orr 2021, viitattu 11.9.2021).

Useista konsolipeleistä on tehty älypuhelimille joko joiltakin ominaisuuksiltaan karsittu versio tai täysin konsoliversiota vastaava mobiilipeli, jonka käyttöliittymä on muokattu matkapuhelimiin sopivaksi. Kymmenen vuotta sitten mobiilipelimarkkinoiden markkina-arvo oli n. 8,6 miljardia dollaria, mutta se on kasvanut huomattavasti ollen vuonna 2020 jo 121,1 miljardia dollaria. Mobiilipelaajien rahankäyttö ylitti vuonna 2019 konsoli- ja PC-pelaajien yhdistetyn rahankäytön. (Liao 2021, viitattu 19.9.2021.)

3 ANDROID

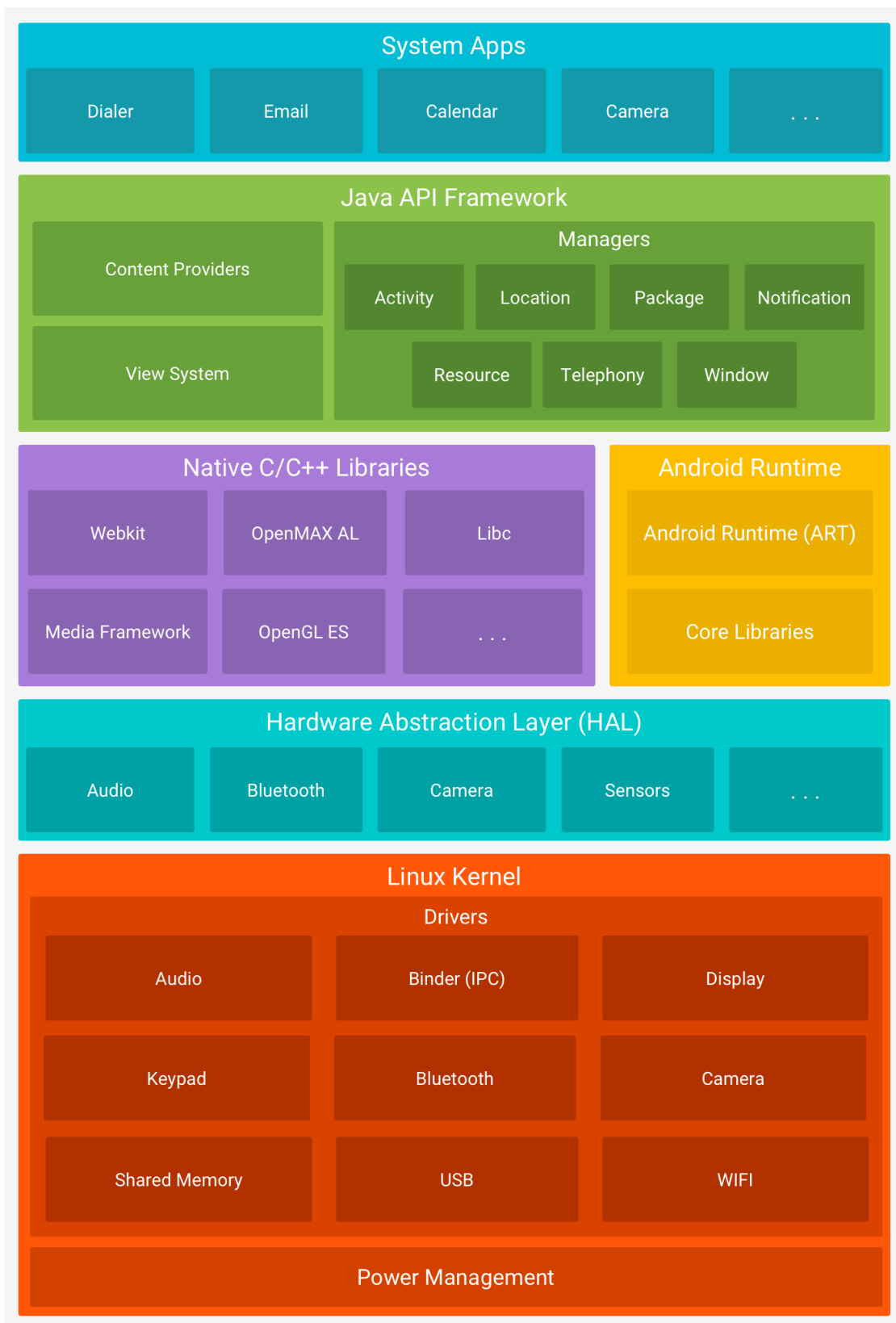
Android on Linux-ytimeen perustuva kosketusnäytöllisille mobiililaitteille suunniteltu ohjelmistopino, joka sisältää käyttöjärjestelmän, väliohjelmistoja ja perusohjelmia käyttäjälle. Android on vapaa ja avoimen lähdekoodin ohjelmisto, mutta Androidin nimi ja logo ovat Googlen tavaramerkkejä. Suurin osa Android-puhelimista käyttääkin Googlen ekosysteemiä, johon kuuluvat mm. sovelluskauppa Play Store, kalenteri- ja ajanhallintaohjelmat ja selainohjelma Chrome. (Hughes 2014, viitattu 24.11.2020.)

3.1 Historia

Androidin kehityksestä vastasi alun perin 2003 perustettu Android Inc., jonka Google osti vuonna 2005 (Thomas 2010, viitattu 20.8.2021). Vuonna 2007 kehitysvastuu siirtyi juuri julkistetulle Open Handset Alliancalle, joka on 84 yrityksen muodostama konsortio. Samalla OHA julkisti Android-sovellusalustan. (Open Handset Alliance 2007, viitattu 20.8.2021.) Ensimmäinen kaupallinen Android-versio 1.0 julkaistiin 23. elokuuta 2008.

3.2 Arkkitehtuuri

Androidin järjestelmäarkkitehtuuri koostuu kuvassa 3 näkyvistä komponenteista. Laitteistoa lähinnä oleva komponentti on järjestelmän ydin (kernel) ja siihen kuuluvat laitteistoajurit. Kuvassa ytimen yläpuolella näkyvä Hardware Abstraction Layer eli HAL tarjoaa ohjelmille rajapinnan laitteistoresurssien käyttöä varten. Seuraavassa kerroksessa ovat C- ja C++-kieliset kirjastot natiivikielellä kirjoitettuja Androidin järjestelmäkomponentteja ja palveluita varten. Näitä ovat mm. HAL ja ART. ART eli Android Runtime on ajoympäristö, joka kääntää ohjelmat natiiville konekielelle. Jokainen sovellus Androidissa ajetaan omassa prosessissaan ja ART-instanssissaan. Java API Framework on Java-kielinen sovelluskehys, jonka ohjelmointirajapintojen avulla sovellukset voivat käyttää kaikkia Androidin ominaisuuksia. Päällimmäisenä kerroksena Androidissa ovat järjestelmäsovellukset, jotka tulevat Androidin mukana. Näitä ovat esimerkiksi sähköposti-, tekstiviesti, kalenteri- ja yhteystietosovellukset. Nämä ovat järjestelmän kannalta täysin normaaleja sovelluksia. Käyttäjä voi halutessaan vaihtaa vaikkapa oletussähköpostisovellukseksi haluamansa kolmannen osapuolen sähköpostisovelluksen. (Google 2021b, viitattu 30.8.2021.)



Kuva 3. Androidin arkkitehtuuri.

3.3 Google Play

Google Play on Googlen omistama digitaalinen sisältöpalvelu. Se on Googlen virallinen sovelluskauppa Android-laitteille. Se toimii myös digitaalisen median ostospaikkana. Google Playstä käyttäjät voivat ladata

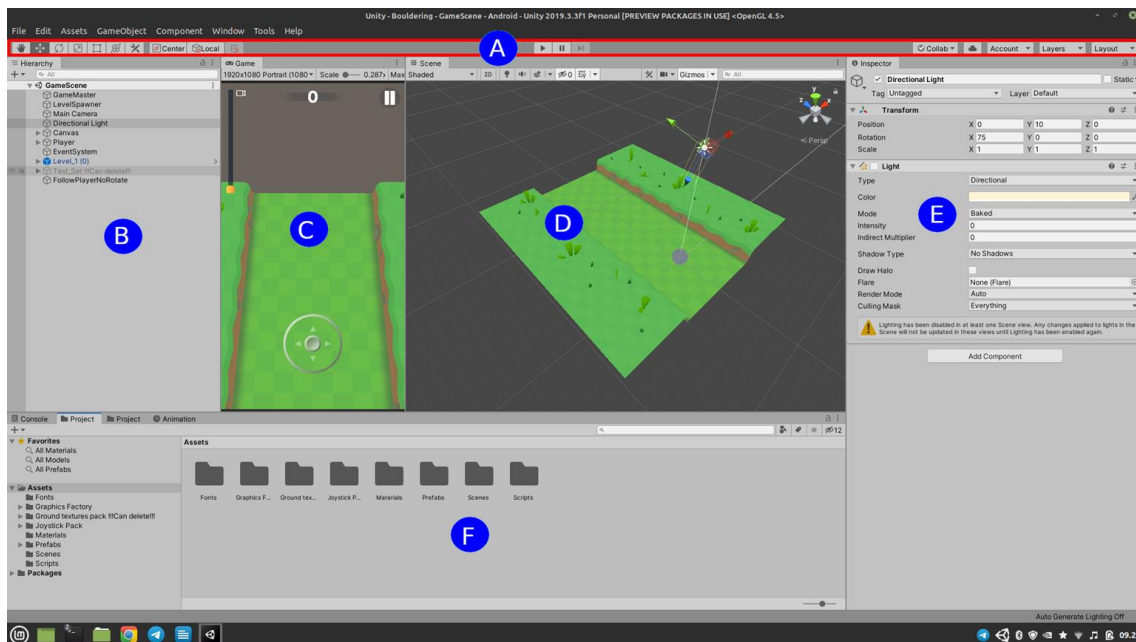
- 1 pelejä
- 2 mobiilisovelluksia
- 3 elokuvia ja tv-ohjelmia
- 4 kirjoja.

Palvelusta ladatut kirjat, elokuvat ja tv-ohjelmat ovat käytettävissä myös selaimella ja iOS-laitteilla. Palveluun kuului aikaisemmin myös Google Play Music, joka oli musiikin ja podcastien kuunteluun tarkoitettu suoratoistopalvelu. (Delfino 2021, viitattu 14.9.2021.)

Google Playn ladatuin peli on battle royale -lajityyppiin kuuluva Garena Free Fire, joka on ladattu yli miljardi kertaa (Pacheco 2021, viitattu 15.9.2021). Muista sovelluksista, pois lukien Googlen omat esiasennetut sovellukset, suosituin on Facebook, jota on ladattu yli viisi miljardia kertaa (Google 2021a, viitattu 14.9.2021). Google Play vastaa sovellusten ja osin myös Android-järjestelmän päivityksistä (Ben Ari & Ghuloum 2019, viitattu 15.9.2021).

4 UNITY

Unity on Unity Technologiesin kehittämä pelien tekemiseen tarkoitettu ohjelmistokehys. Unity julkaistiin alun perin vuonna 2005 Mac OS X -pelien kehittämistä varten. Vuonna 2018 Unityllä pystyi kehittämään pelejä jo yli 25 eri alustalle mukaan lukien Android, iOS, Windows, macOS, PlayStation 4 ja useat VR-alustat (Unity 2021, viitattu 16.8.2021).



Kuva 4. Unityn käyttöliittymä

4.1 Käyttöliittymä

Kuvassa 4 näkyy Unityn käyttöliittymän perusnäkymä. Käyttäjä voi halutessaan vaihdella eri komponenttien paikkaa mieleisekseen.

- A Työkalurivillä on tärkeimmät käytettävät ominaisuudet. Vasemmalla näkyy peliohjainten (GameObjects) ja Scenen manipulointiin tarkoitettuja työkaluja. Keskellä ovat scenen käynnistys- (Play), tauotus- (Pause) ja askel- (Step) napit ja Unity-tunnukseen ja editorin asetteluun liittyviä toimintoja on työkalurivin oikeassa reunassa.
- B Hierarkiaikkunassa näkyvät kaikki Scenen peliohjeet ja niiden suhteet toisiinsa.
- C Peli-ikkunassa näkyy simuloituna, miltä peli näyttäisi. Simulaatio käynnistetään käynnistysnapista.
- D Scene-ikkunassa voidaan tarkastella ja muokata valittua sceneä.

- E Oikealla näkyvässä Inspector-ikkunassa näkyy valitun peliobjektin komponentit ja niiden ominaisuudet. Ikkunassa voidaan lisätä komponentteja ja muokata niiden ominaisuuksia.
- F Projekti-ikkunassa näkyy kaikki projektissa käytössä olevat assetit.

4.2 Unityn käyttäminen

Unityn käyttäminen pelin tekemiseen aloitetaan projektin luomisella. Projektin luomista varten Unity tarjoaa erilaisia projektimalleja erityyppisten pelien tekemiseen. Näitä ovat mm. 2D- ja 3D-mallit. Ne eivät rajoita sitä, millaista peliä tai sovellusta projektissa voi tehdä. Ne tarjoavat vain valmiita asetuksia, jotka helpottavat tiettytyyppisen pelin tekemistä. (Unity 2017a, viitattu 20.8.2021.)

Kaikki pelinsisäinen toiminta sisältyy sceneen. Sceneä voidaan ajatella yhtenä pelin kenttänä tai tasona. Pienissä peleissä koko peli voi olla yhdessä scenessä. Pelit rakennetaan lisäämällä sceneihin peliobjekteja (GameObject) ja määrittelemällä niiden ominaisuudet ja toiminnallisuudet.

Kaikki mitä peliin halutaan lisätä ovat peliobjekteja: pelaajahahmo, viholliset, kamerat, maasto, valikot yms. Tyhjä peliobjekti sisältää vain Transform-komponentin, joka määrittelee peliobjektin sijainnin, asennon ja skaalauksen. Eri komponentteja lisäämällä peliobjektille voidaan määritellä mm. 3d-malli, fysiikkasimulaatio, törmäysten tunnistaminen ja reagointi pelaajakontrolleihin. Peliobjekteja voidaan käyttää myös kansioina, joihin voi lisätä muita peliobjekteja. Tämä tapahtuu vetämällä peliobjekti toisen päälle hierarkiaikkunassa. Vedetystä peliobjektista tulee näin toisen peliobjektin lapsiobjekti. Vanhemman sijaintia tai rotaatiota muutettaessa myös lapsiobjekti liikkuu ja kääntyy. (Unity 2017b, viitattu 20.8.2021.)

Vetämällä peliobjekti hierarkiaikkunasta projekti-ikkunaan siitä tehdään prefab-asetti, joka sisältää peliobjektin ja sen lapsiobjektit kaikkine komponentteineen ja niiden arvoineen. Prefab-asettiin tehdyt muutokset tulevat voimaan kaikissa siitä tehdyissä instansseissa. Prefab-assetit toimivat malleina, joiden mukaan sceneen voidaan luoda prefab-instansseja joko suoraan editorissa vetämällä malli scene-ikkunaan tai käyttämällä Instantiate-metodia jossakin scenessä olevan peliobjektin skriptissä. Prefabit helpottavat siis useiden samanlaisten peliobjektien luomista ja muokkaamista. (Unity 2018, viitattu 20.8.2021.)

Unityn skriptit (Scripts) ovat C#:lla kirjoitettuja ohjelmanpätkiä, joilla voidaan lisätä sceneen käyttäjän määrittelemiä toimintoja. Niiden avulla voidaan esimerkiksi saada pelaajahahmo reagoimaan

kontrolleihin tai määrittellä miten viholliset liikkuvat tai mitä tapahtuu, jos pelaaja törmää esteeseen. Jotta skripti ajettaisiin peliä pelatessa, se pitää lisätä komponentiksi peliobjektiin ja tämä peliobjekti pitää lisätä sceneen. Yleensä skripti lisätään siihen peliobjektiin, jonka ominaisuuksiin ja toimintaan halutaan vaikuttaa, mutta skriptit voivat vaikuttaa myös muihin scenessä oleviin peliobjekteihin.

Peliä voidaan testata Unityn Game-ikkunassa. Tässä ikkunassa näkyy, miltä peli näyttäisi, jos sitä pelattaisiin esim. älypuhelimessa. Peli käynnistyy painamalla työkalurivin Play-nappia. Myös Scene-ikkunassa näkyy miten peliobjektit käyttäytyvät pelin aikana.

Jotta peliä voisi testata puhelimessa, siitä pitää tehdä koontiversio. Tämä tapahtuu Unityssä valitsemalla File-valikosta Build Settings, joista valitaan scenet, jotka lisätään koontiversioon, alusta jolle koontiversio tehdään ja muut tarvittavat asetukset. Asetusten valitsemisen jälkeen painetaan Build-nappia, jonka jälkeen koontiversiolle hyväksytään tiedostonimi. Tiedoston päätteeksi tulee apk, joka on lyhenne sanoista Android Package. Se sisältää ohjelman ja tarvittavat data- ja resursitiedostot. Unityn tehtyä koontiversion se voidaan siirtää puhelimelle ja asentaa siihen testaamista varten.

5 PELIN TOTEUTUS

Luvussa 5.1 kuvataan ensin lyhyesti pelin idea. Tämän jälkeen luvussa 5.2 käydään läpi pelin toteutusta pääpiirteittäin.

5.1 Pelin idea

Pelin alkuperäisenä ideana oli vieriä pyöreällä kivenjärkeellä eteenpäin väistellen samalla kuoppia ja litistään vihollisia. Litistetyt viholliset tarttuisivat järkäleeseen tehden siitä painavamman, mikä vaikeuttaisi järkäleen ohjaamista. Kuoppien sijaan peliin päätettiin tehdä erilaisia maanpäällisiä esteitä, koska näin maa voitiin tehdä yhtenä kappaleena ja esteiden luominen yksinkertaistui. Vihollisten liiskaaminen muutettiin esineiden keräämiseksi. Kerätyistä esineistä saa pisteitä, mutta järkäleen paino ei muutu. Tarkoituksena oli, että peliin lisättäisiin myös esineitä, jotka antaisivat pelaajalle erityiskykyjä, joilla voisi esimerkiksi tuhota esteitä, mutta näitä ei ehditty lisätä peliin. Myös kenttäeditoria suunniteltiin, mutta sekin jäi pois liian työlääksi. Pelissä on nykyisellään vain yksi loputon kenttä, joka generoidaan satunnaisesti.

5.2 Pelin toteutus

Pelikentän pohja ja sivut tehtiin ensin Unityn editorilla venyttämällä ja asettelemalla yksinkertaisia 3D-objekteja, kuten kuutio ja taso. Näin saadusta pelikentän palasta tehtiin prefab-asetti, jotta uusia palasia voitaisiin lisätä skriptassa ohjelmallisesti pelaajan edetessä kentällä.

Kentän luomisesta huolehtii LevelSpawner-skripti, joka luo uuden palan kenttää aina pelaajan liikkuessa eteenpäin niin, että uusi pala ilmestyy aina pelaajan näkökentän ulkopuolelle. Skripti huolehtii myös ohitettujen palasten poistamisen scenestä. Luoduille palasille arvotaan sen jälkeen esteet ja kerättävät esineet. Pelaajalle jätetään aina reitti kentän läpi.

Pelihahmona toimivan järkäleen toiminnot on määritelty Player-skriptassa. Pelaajan järkälettä ohjataan virtuaaliohjaimella. Asset Storesta löytyi tätä varten ilmainen Joystick Pack -asettipaketti. Paketissa on virtuaaliohjaimen näyttämistä varten kuvat sekä skripti, joka palauttaa virtuaaliohjaimen vaaka- ja pystysuuntaisen liikkeen liukulukuna -1:n ja 1:n välillä. Ohjainta liikuttaessa sivusuunnassa järkäleeseen kohdistetaan liikkeen suuruutta vastaava voima ja Unityn fysiikkamoottori

huolehtii voiman aiheuttamasta järkäleen suunnan muuttumisesta. Player-skriptassa on myös määritelty, mitä tapahtuu, jos järkäle osuu esteeseen tai kerättävään esineeseen.

Pelin tilasta huolehtii GameMaster-skripti. Pelin eri tiloja ovat mm. MainMenu, Play ja Paused. Pelitilasta riippuu mitä ruudulla ja valikoissa näkyy ja onko peli pysäytetty vai eteneekö se. Kts. kuvat 5 ja 6. GameMaster-skriptin pohjana on käytetty skriptiä, jota on käytetty useissa Lost Hoodie Oy:n peleissä, joissa on sama pelitiloihin perustuva toimintalogiikka ja valikkorakenne. GameMasterissa on myös määritelty mainosten näyttäminen ja analytiikkatietojen kerääminen, jotka myöskin toimivat samalla tavalla yrityksen eri peleissä.

```
// For handling Game States
public enum GameState
{
    MainMenu,
    Play,
    Paused,
    End,
    Continue,
    Null,
    //Tutorial
}
```

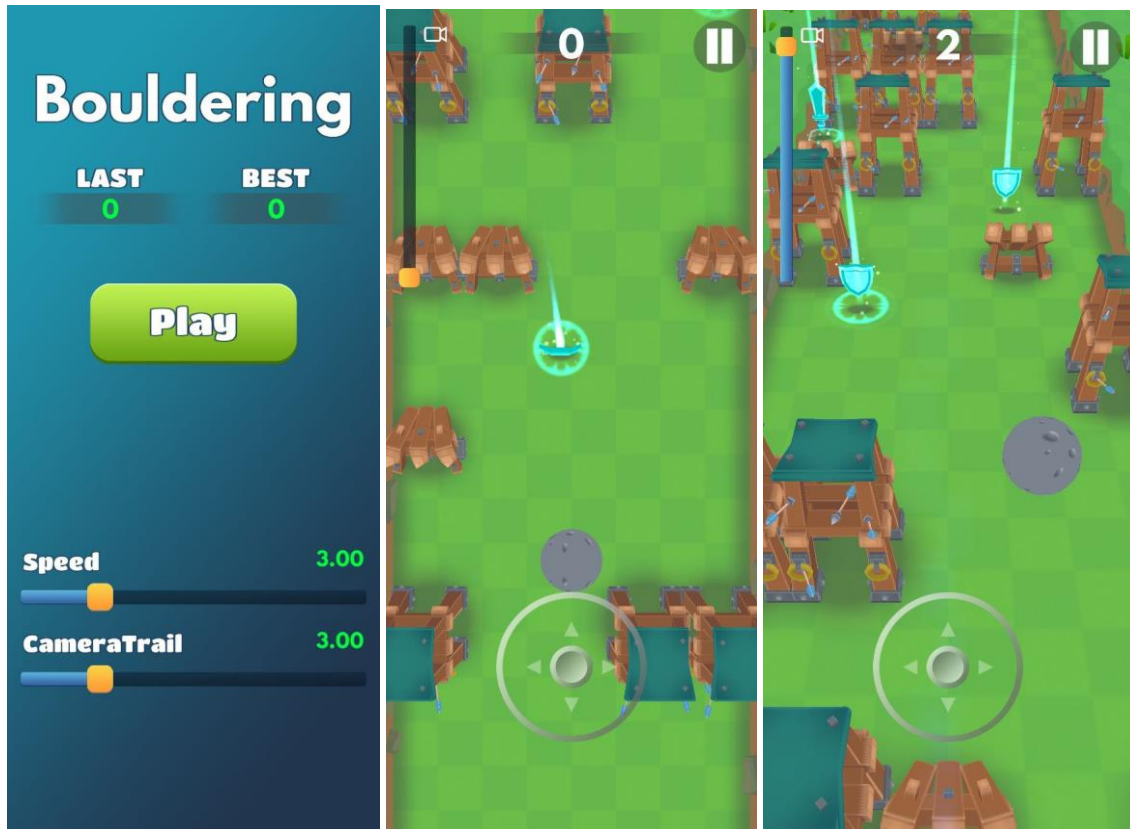
Kuva 5. GameMaster-skriptassa määritellyt pelin mahdolliset tilat.

```
switch (GameStateNow)
{
    case GameState.MainMenu:
        InMenu();
        ui_Play.SetActive(true);
        ui_Pause.SetActive(false);
        // Not active
        ui_Restart.SetActive(false);
        ui_Resume.SetActive(false);
        ui_Pause.SetActive(false);
        ui_GameOver.SetActive(false);
        //ui_Continue.SetActive(false);
        break;
    case GameState.Play:
        // Ad stuff
        //AdnalyticMaster.HideBanneri();
        // Active
        ui_Pause.SetActive(true);
        ui_CameraAngle.SetActive(true);
        // Not active
        ui_Menu.SetActive(false);
        //ui_Continue.SetActive(false);
        break;
```

Kuva 6. Osa switch-komennosta, joka määrittelee mitä elementtejä ruudulla näytetään missäkin pelitilassa.

Edellä mainittujen tehtävien lisäksi GameMaster-skripti huolehtii pelin pistelaskusta.

5.3 Pelin graafinen ilme



Kuvat 7, 8 ja 9. Pelin alunäyttö ja pelikuvaa kameran eri kallistuskulmissa.

Kuvassa 7 näkyy pelin alunäyttö. Kuvassa ylipänä on pelin nimi Bouldering. Sen alla ovat edellisen pelin pisteet ja korkein pelissä saavutettu pistemäärä. Ruudun keskipaikkeilla on pelin käynnistysnappi. Ruudun alaosassa on pelin pelattavuuteen liittyviä liukusäätimiä, joita ei ole tarkoitus näyttää pelin lopullisessa versiossa. Speed-säätimellä muutetaan pallon pyörimisnopeutta. CameraTrail-säätimellä muutetaan kameran etäisyyttä pallost.

Kuvat 8 ja 9 on otettu pelin aikana kameran kuvatessa peliä suoraan ylhäältäpäin ja yläviistosta takaapäin. Vasemmalla ylhäällä näkyy liukusäädin, jolla kameran kulmaa voidaan säätää. Keskellä ylhäällä on pelin sen hetkiset pisteet. Oikealla ylhäällä on nappi, josta pelin voi keskeyttää. Alhaalla näkyy virtuaaliohjain, jolla palloa kontrolloidaan. Muu kuvissa näkyvä on pelikentän maastoa, kerättäviä esineitä ja väisteltäviä esteitä sekä pelissä ohjattava järkäle.

Esteet ja kerättävät esineet tehtiin aluksi Unityn editorissa yksinkertaisina kuutioina, joiden koko muokattiin kenttään sopivaksi. Näin peliä pystyi kehittämään ja testaamaan samalla, kun Lost Hoodie Oy:n graafikko suunnitteli lopulliset versiot pelin kenttäpaloista, esteistä ja esineistä.

5.4 LevelSpawner-toiminnallisuus

Suurin osa pelin ohjelmointiin kuluneesta ajasta kului LevelSpawner-skriptan toimintoihin. Etenkin sen varmistaminen, että pelaajalle jäi aina mahdollinen reitti pelikentän läpi, oli haastavaa. Ongelmia aiheutti mm. se, että vaikka eräs algoritmi pystyi varmistamaan sen, että pelikentän läpi kulki aina vähintään yksi mahdollinen reitti, pelaaja pystyi valitsemaan myös toisen reitin, joka päättyisi umpikujaan myöhemmin. Ongelman ratkaisu oli kuitenkin lopulta yksinkertainen. Pelikentälle arvottiin satunnaisesti esteitä, jonka jälkeen niiden läpi tehtiin reitti, jolta poistettiin esteet.

```
// clearlane is a path with no obstacles
int clearLane = Random.Range(0, lanes);
for (int row = 0; row < 9; row += 2)
{
    int obstaclesPerRow = Random.Range(minObstacles, maxObstacles+1);
    for (int obstacleNumber = 0; obstacleNumber < obstaclesPerRow; obstacleNumber++)
    {
        int randomLane = Random.Range(0, lanes);
        int safety = 0;

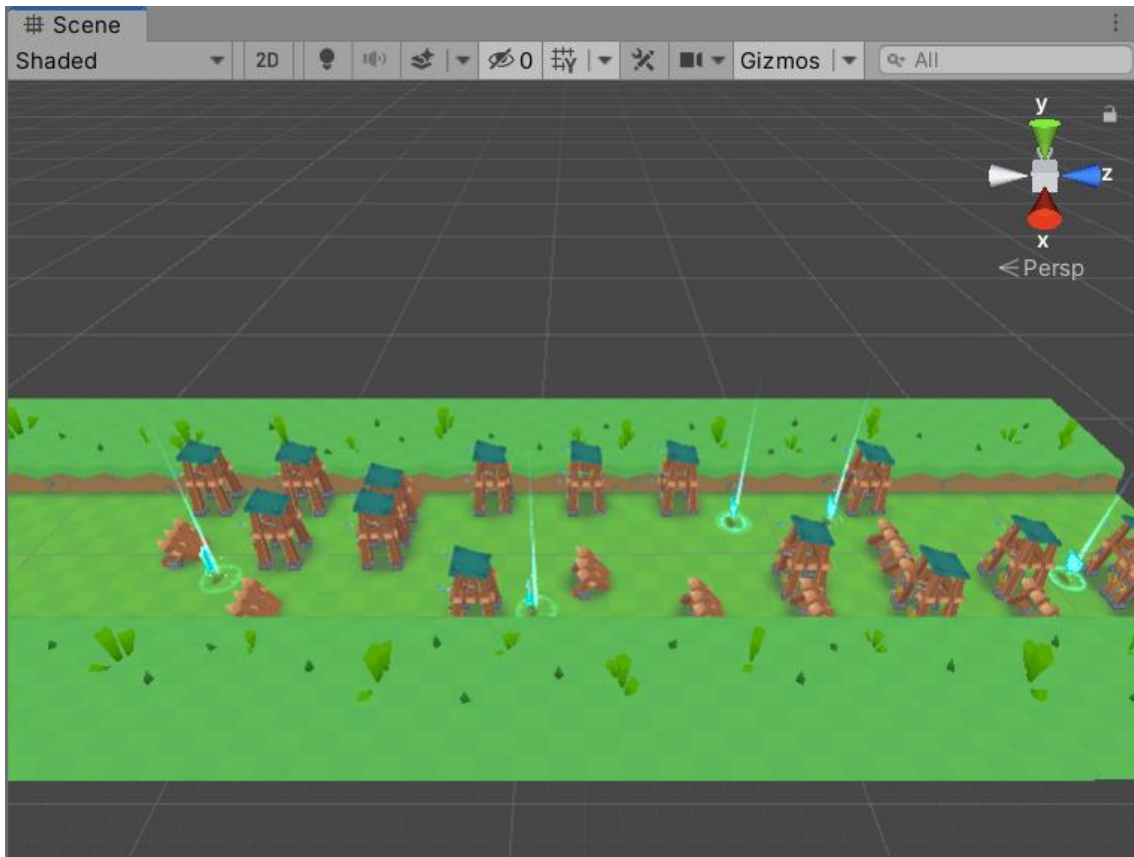
        while (obstacleGrid[randomLane, row] != 0 && safety < 3)
        {
            randomLane = Random.Range(0, lanes);
            safety++;
        }
        obstacleGrid[randomLane, row] = 1;
    }

    // Cut a path for the player through the obstacleGrid, clearing first the lane that it landed in previous row.
    obstacleGrid[clearLane, row] = 0;

    // clearLane moves randomly one way or another in the grid staying inside the grid.
    clearLane += Random.Range(-1, 2);
    clearLane = Mathf.Clamp(clearLane, 0, lanes - 1);
    // Clears any possible obstacles on the obstacleGrid.
    obstacleGrid[clearLane, row] = 0;
}
```

Kuva 10. LevelSpawner-skriptasta osa, joka arpoo kentälle esteet ja puhdistaa reitin pelaajalle.

Suunnittelupalaverissa päätettiin myöhemmin, että mahdolliset esteet ja kerättävät esineet tulisivat vuorottaisille riveille, kuten kuvassa 11 näkyy. Päätös liittyi pelin graafisen ilmeen muokkaamiseen. Tämä ratkaisi kuitenkin samalla ongelman umpikujaan joutumisesta.



Kuva 11. Kuvakaappaus Scene-ikkunasta, jossa näkyy esteiden ja kerättävien esineiden vuorottelevat rivit.

6 POHDINTA

Tässä opinnäytetyössä käytiin läpi mobiilipelin toteutus Unityllä. Lukuun ottamatta GameMaster-skriptan pohjaa ja virtuaaliohjaimen asettipakettia kaikki kehitystyö peliä varten tehtiin ilman aiempaa materiaalia.

Pelistä saatiin tehtyä toimiva versio, jossa oli lähes kaikki ominaisuudet, jotka olivat alkuperäisessä suunnitelmassa. Peliin suunniteltiin kehitystyön aikana lisäominaisuuksia, kuten kenttäeditori, kerättävistä esineistä saatavat erikoiskyvyt ja monipuolisemmat kentät. Näitä ei kuitenkaan pystytty toteuttamaan, koska ne olisivat viivästyttäneet liikaa projektia. Pelin kehityksessä tuli kuitenkin lähes vuoden mittainen tauko muista opinnäytetyöhön liittymättömistä syistä ja Lost Hoodie keskeytti toimintansa sinä aikana. Play Store -integraatio, Unity- ja Google-mainokset sekä analytiikkatietojen keräys jouduttiin jättämään pois, koska tarvittavia Google- ja Unity-tunnuksia ei ollut enää käytössä. Peliin tuli myös ominaisuuksia, joita ei ollut suunniteltu alun perin. Kamerakulman säätö oli yksi niistä. Toinen lisätty ominaisuus oli pelattavan järkäleen nopeuden lisääminen virtuaaliohjaimesta eteenpäin työntämällä.

Suurin ongelma lopullisen pelin kannalta on se, että se ei ole tällaisenaan tarpeeksi kiinnostava. Jos kaikki suunnitellut ominaisuudet olisi pystytty lisäämään peliin, se olisi ollut huomattavasti mielenkiintoisempi. Jos Lost Hoodie jatkaa vielä toimintaansa, voidaan pelin kehitystä vielä jatkaa ja peliin voidaan lisätä suunnitellut ominaisuudet.

Pelin julkaiseminen Googlen Play Storessa ei vaadi juurikaan investointeja. Tämän takia kilpailu on erittäin kovaa ja joukosta erottuminen on vaikeaa. Peliala on erittäin haastava runsaan kilpailun takia. Pienissä yrityksissä jokaisen tekijän rooli on iso, koska lisää työvoimaa ei voida palkata ennen kuin yritys saa peleistään tarpeeksi tuottoa tai muuta kautta lisärahoitusta toimintaansa. Tämän takia peliä suunniteltaessa ja toteutettaessa piti miettiä lähes kaikkia pelikehityksen osa-alueita. Opinnäytetyötä tehdessä tietotaito ja kokemus mobiilipelien kehittamisestä kasvoi huomattavasti. Samalla kertyi kokemusta monialaisessa tiimissä työskentelystä. Opittuja tietoja ja taitoja voidaan hyvin käyttää jatkossa muiden mobiilipelien kehittämisessä ammatillisessa ympäristössä.

LÄHTEET

APS News. 2008. October 1958: Physicist Invents First Video Game. This Month in Physics History. October 2008 (Volume 17, Number 9). Viitattu 24.10.2020, <https://www.aps.org/publications/apsnews/200810/physicshistory.cfm>.

Baer, R. 2005. Videogames: In the Beginning. Springfield: Rolenta Press.

Ben Ari, M & Ghuloum, A. 2019. Fresher OS with Projects Treble and Mainline. Android Developers Blog. Viitattu 15.9.2021, <https://android-developers.googleblog.com/2019/05/fresher-os-with-projects-treble-and-mainline.html>.

Bonnington, C. 2013. 5 Years On, the App Store Has Forever Changed the Face of Software. Wired. Viitattu 25.8.2021, <https://www.wired.com/2013/07/five-years-of-the-app-store/>.

Delfino, D. 2021. What is Google Play? The online store for Android devices, explained. Business Insider. Viitattu 14.9.2021, <https://www.businessinsider.com/what-is-google-play>.

Hughes, T. 2014. Google and Android Are Not the Same...and That's a Good Thing. App Developer Magazine. Viitattu 24.11.2020, <https://appdeveloper magazine.com/google-and-android-are-not-the-same...-and-that%E2%80%99s-a-good-thing/>.

Google. 2021a. Facebook. Google Play. Viitattu 14.9.2021, <https://play.google.com/store/apps/details?id=com.facebook.katana>.

Google. 2021b. Platform Architecture. Viitattu 30.8.2021, <https://developer.android.com/guide/platform>.

Koerper, H. & Whitney-Desautels N. 1999. Astragalus Bones: Artifacts Or Ecofacts? Pacific coast Archaeological Society Quarterly. Volume 35. Numbers 2 & 3.

Liao, S. 2021. Forget next-gen consoles. The biggest gaming platform is already in your pocket. The Seattle Times. Viitattu 19.9.2021, <https://www.seattletimes.com/business/technology/forget-next-gen-consoles-the-biggest-gaming-platform-is-already-in-your-pocket/>.

Open Handset Alliance. 2007. Industry Leaders Announce Open Platform for Mobile Devices. Viitattu 20.8.2021, http://www.openhandsetalliance.com/press_110507.html.

- Orr, A. 2021. Top 10 highest-grossing mobile games of all time. Wired. Viitattu 11.9.2021 <https://www.pocketgamer.biz/feature/77017/top-10-highest-grossing-mobile-games-of-all-time/list/>.
- Manninen, T. 2007. Pelisuunnittelijan käsikirja – Ideasta eteenpäin. Oulu: Rajalla.
- Pacheco, S. 2021. Garena Free Fire Crosses 1 Billion Downloads On The Google Play Store. Digit. Viitattu 15.9.2021, <https://www.digit.in/news/gaming/garena-free-fire-1-billion-downloads-google-play-store-60241.html>.
- Thomas, O. 2010 Google exec: Android was “best deal ever”. VentureBeat. Viitattu 20.8.2021, <https://venturebeat.com/2010/10/27/google-exec-android-was-best-deal-ever/>.
- Unity. 2017a. 2D or 3D projects. Unity. Viitattu 20.8.2021, <https://docs.unity3d.com/560/Documentation/Manual/2Dor3D.html>.
- Unity. 2017b. GameObjects. Unity. Viitattu 20.8.2021, <https://docs.unity3d.com/Manual/GameObjects.html>.
- Unity. 2018. Prefabs. Unity. Viitattu 20.8.2021, <https://docs.unity3d.com/Manual/Prefabs.html>.
- Unity. 2021. Build once, deploy anywhere. Unity. Viitattu 16.8.2021, <https://unity.com/features/multiplatform>.
- Vogelstein, F. 2008. The Untold Story: How the iPhone Blew Up the Wireless Industry. Wired. Viitattu 22.2.2021, <https://www.wired.com/2008/01/ff-iphone/>.
- Wallach, O. 2020. 50 Years of Gaming History, by Revenue Stream (1970-2020). Viitattu 13.8.2021, <https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/>.
- Wright, C. 2016a. A Brief History of Mobile Games: In the beginning, there was Snake. Viitattu 16.8.2021, <https://www.pocketgamer.biz/feature/10619/a-brief-history-of-mobile-games-in-the-beginning-there-was-snake/>.
- Wright, C. 2016b. A Brief History of Mobile Games: 2002 - Wake up and smell the coffee. Viitattu 17.8.2021, <https://www.pocketgamer.biz/feature/10705/a-brief-history-of-mobile-games-2002-wake-up-and-smell-the-coffee/>.