

Joona Leskinen

SUMMER SPORTS -MOBIILPELI

SUMMER SPORTS -MOBIILPELI

Joona Leskinen
Opinnäytetyö
Syksy 2015
Tietotekniikan koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehityksen sv

Tekijä: Joonas Leskinen

Opinnäytetyön nimi: Summer Sports -mobiilipeli

Työn ohjaaja: Pertti Heikkilä

Työn valmistuslukukausi- ja vuosi: Syksy 2015

Sivumäärä: 29

Työn tavoitteena oli luoda mobiilipeli, jossa tarkoituksena pelata yksinkertaisia kesälajeja ja syöttää tulokset tuloslistalle.

Kehityksessä käytettiin Unity 3D -kehitysympäristöä sovelluksen kehitykseen sekä Gimp-kuvankäsittelyohjelmistoa grafiikkojen luontiin. Projektissa käytettiin joitain avoimen lähdekoodin valmiita komponentteja AdWords-rajapinnan käyttöön sekä Google Play Servicen käyttöön.

Suurin osa tavoitelluista ominaisuuksista saatiin kehitettyä ajoissa sekä testattua pienellä testaajaryhmällä. Lopputuloksena saatiin hyvin toimiva prototyyppi mobiilipelistä, jossa mukana oli mainokset sekä Google-kirjautuminen.

Peliä ei vielä julkaistu Play-kaupassa, vaan se vaatii vielä paljon kehittämistä, että se olisi julkaisukunnossa.

Asiasanat: Unity 3D, Gimp, mobiilipelit, Canvas, Android

ABSTRACT

Oulu University of Applied Sciences
Information technology

Author(s): Joonas Leskinen

Title of thesis: Summer Sports Mobile game

Supervisor(s): Pertti Heikkilä

Term and year when the thesis was submitted: Autumn 2015 Number of pages: 29

The goal of the thesis was to design and develop a 2D-mobile game for Android, which contains few different summer sport games and ability to upload results to Google play service leaderboard.

Game was developed with Unity 3D engine and graphics was made with Gimp. Open source Google Play service and AdWords-plugins were used to ease access to those APIs.

Most of the desired features were created in time and mostly tested with small tester group. As a result, a well-functioning prototype of a mobile game, which contained Google-login and test advertisements.

Game is not yet published in the play store, because it still requires a lot of development so that it would be in condition to publish for all.

Keywords: Unity 3D, mobile games, Android, Gimp, Canvas

SISÄLLYS

TIIVISTELMÄ.....	3
ABSTRACT.....	4
SISÄLLYS.....	5
1 JOHDANTO.....	6
2 ANDROID.....	7
3 UNITY 3D.....	8
3.1 Historia.....	9
3.2 Ohjelmointi Unity 3D:ssä.....	9
3.3 MonoBehaviour.....	9
3.4 Animaattori.....	10
4 UNITY CANVAS.....	11
4.1 Canvaksen komponentit.....	11
4.2 Ulkoasun asetukset.....	14
5 GOOGLE ADWORDS.....	17
6 GOOGLE PLAY SERVICE.....	18
6.1 Google Play Service Unity 3D:ssä.....	18
6.2 Saavutukset.....	19
6.3 Tulostaulut.....	19
7 PELIN SUUNNITTELU.....	20
8 PELIN TOTEUTUS.....	21
8.1 Juoksu.....	21
8.2 Keihäänheitto.....	21
8.3 100 m:n juoksu.....	22
8.4 110 m:n aitajuoksu.....	23
8.5 Jousiammunta.....	23
8.6 Hahmon animointi.....	24
8.7 Päävalikko.....	25
9 YHTEENVETO.....	28
LÄHTEET.....	29

1 JOHDANTO

Mobiilipelimarkkinat ovat nopeasti kasvava ala. On yksittäisiä tähdenlentoja, jotka saavuttavat suuren suosion, sekä kovalla työllä tehtyjä pelejä, jotka eivät välttämättä saa minkäänlaista huomiota. Tästä huolimatta mobiilipelin kehittäminen voi olla todella kannattavaa, ideasta huolimatta. Hyvänä esimerkkinä voidaan sanoa vietnamilainen Flappy Bird. Flappy Bird on graafisesti ja ohjelmallisesti hyvin yksinkertainen, mutta silti parhaimpina päivinä peli tuotti jopa 50 000 € päivässä. (1.)

Tässä projektissa luotiin vanhan Nintendo Entertainment Systemin (NES) pelikonsolin Track and Field -pelin tapainen mobiilipeli Android-mobiililaitteille. Pelissä on ideana pelata erilaisia urheilulajeja, jotka vaativat nopeutta. Pelin eri lajeja ovat keihäänheitto, 100 m:n juoksu, 110 m:n aidat ja jousiammunta.

Ensimmäisenä peliä suunnitellessa piti valita pelimoottori, jolla projekti toteutettaisiin. Paras vaihtoehto olisi tehdä oma pelimoottori käyttäen Android SDK:ta, jolloin koko projekti luotaisiin tyhjästä. Aikataulun vuoksi tämä ei ollut kovin kannattava vaihtoehto eikä projektista todennäköisesti olisi tullut puoliakaan valmiiksi tavoiteajassa.

Unreal Engine 4 oli toinen vaihtoehto sen monialustaisuuden takia. Pelimoottorina Unreal Engine 4 on monipuolinen ja graafisesti kaunis. Unreal Engine 4:n hienoja ominaisuuksia on muun muassa sen pelimoottorin avoimuus, joka mahdollistaa moottorin lähdekoodien muuttamisen omaan tarpeeseen sopivaksi. Tähän projektiin Unreal Engine 4 olisi ollut kuitenkin liian massiivinen.

Lopulta päädyin kuitenkin Unity 3D -pelimoottoriin sen helppouden ja monipuolisuuden takia.

Unity 3D oli valmiiksi tuttu ympäristö, joten tälle alustalle olisi hyvä lähteä luomaan peli aikarajojen puitteissa.

2 ANDROID

Android on Android Inc yhtiön kehittämä ilmainen mobiilikäyttöjärjestelmä, jonka Google myöhemmin osti. Androidin ensimmäinen versio julkistettiin 5.11.2007 ja se nousi älypuhelinien markkinojohtajaksi vuonna 2010. Android perustuu avoimen lähdekoodin GPLv2-lisensioituun Linux-käyttöjärjestelmäyttimeen. Vuonna 2011 Androidin markkinaosuus oli 52,5 % ja vuonna 2013 jopa 70 %, joten mobiilipelimarkkinat ovat hyvin laajat. (2.)

Androidia on moitittu sen sirpaleisuuden takia. Android-laitteita on monilla eri näyttö vaihtoehdoilla ja käyttöjärjestelmäversioilla. Tämä tekee Android-sovellusten kehittämisestä ja testaamisesta hie-man haastavampaa. Sirpaleisuuden takia moni kehittääkin ensimmäisen version sovelluksesta Ap-plen iOS-alustalle. (2.)

3 UNITY 3D

Unity 3D on Unity Technologiesin kehittämä pelimoottori ja kehitysympäristö. Unity 3D:n sisäisellä kehitysympäristöllä on erittäin helppo kehittää pelejä monille eri alustoille. Unity 3D:llä pystyy yhden projektin kääntämään helposti monelle eri laitteelle, tekemättä juurikaan muutoksia lähdekoodeihin.

Alla listattu kaikki Unity 3D:n tukemat alustat.

1. iOS (Applen mobiilikäyttöjärjestelmä)
2. Android
3. Windows Phone 8
4. BlackBerry 10
5. Windows
6. Linux
7. OS X (Applen työpöytäkäyttöjärjestelmä)
8. Playstation 3
9. Xbox 360
10. Wii U
11. Verkkoselaimet

Unity 3D:n grafiikkamoottori tukee useita eri grafiikkarajapintoja mahdollistaen monialustaisuuden. Direct 3D on Microsoftin suljettu rajapinta ja sitä käytetään vain Windows-, Windows Phone- ja Xbox-alustoilla. Direct 3D mahdollistaa grafiikan manipuloinnin laitteistotasolla käyttämällä laitteistokiihdytystä ja on tästä syystä hyvin tehokas. OpenGL on avoimen lähdekoodin grafiikkarajapinta ja sitä käytetään Mac-, Linux- ja Playstation 3 -järjestelmissä. OpenGL:ää voi myös käyttää Windows-pohjaisissa sovelluksissa. Unity 3D tukee useimpia suosittuja grafiikkatyyppejä, muun muassa .FBX-, .3DS-, .max-, .obj- ja .blend-tiedostoja. (3.)

3.1 Historia

Unity 3D:n kehitys alkoi vuonna 2001 ja julkaistiin ensimmäisen kerran vuonna 2005. Alun perin Unity 3D kehitettiin pelien kehitykseen Applen OS X -ympäristöön. Vuonna 2007 Unity 3D:stä julkaistiin versio 2.0 ja sitä kehittävä Unity Technologies oli kasvanut monikertaiseksi alkuperäisestä koostaan. Vuonna 2010 Unity 3D:stä julkaistiin versio 3.0 ja moottori alkoi olemaan huomattavasti monipuolisempi. Vuonna 2012 Unity 3D:stä julkaistiin versio 4.0 ja myöhemmin vuonna 2014 julkaistiin vielä versio 4.6, joka toi mukanaan uuden Canvas-järjestelmän. Vuonna 2015 julkaistiin Unity 3D 5, joka sisälsi paljon uusia ominaisuuksia 2D-pelien luomiseen, animaatioihin sekä verkotekniikkaan. (3.)

3.2 Ohjelmointi Unity 3D:ssä

Unity 3D:n mukana toimitetaan MonoDevelop-sovelluskehitysympäristö, jolla voidaan luoda kaikki Unity 3D -sovellukseen liitettävät skripti-tiedostot. Unity 3D tukee myös Microsoftin Visual Studio -kehitysympäristöä, jossa on huomattavasti paremmat ominaisuudet koodin kirjoittamiseen. Visual Studiossa on erittäin hyvä Intellisense, joka helpottaa koodin kirjoittamista. Visual Studio ei tosin tue kuin C#-kieltä ilman erillisiä lisäosia.

3.3 MonoBehaviour

Unity 3D:n kooditiedostot periytyvät MonoBehaviour-luokasta. MonoBehaviour on Unity 3D -peliohjainten luokka, joka sisältää valmiina Update, FixedUpdate, Start, Awake ja monta muuta eri funktiota. Näistä kaikki ei ole valmiina näkyvissä, vaan ne pitää itse erikseen kirjoittaa. Jokaiseen peliohjektiin, jota halutaan ohjelmallisesti ohjata jollain tapaa, tulee lisätä skripti, joka periytyy tästä luokasta.

3.4 Animaattori

Unity 3D:ssä on valmiina erittäin monipuolinen animaattori, jolla on mahdollista luoda erilaisia animaatioita kaikkiin peliobjekteihin. Animaattorin käyttö on yksinkertaista, varsinkin kun tehdään kaksiulotteisia animaatioita. Kun uutta animaatiota aletaan luomaan, tulee animaation aikajana näkyviin. Aikajanalla määritetään key frame't, jotka määrittävät kiinteät pisteet, jossa objekti on valitulla hetkellä. Unity 3D:n animaattori ajastaa kaiken näiden pisteiden välillä. Jos objekti ruudussa 0 on sijainnissa 1,0 ja ruudussa 10 sijainnissa 2,0, animaattori liikuttaa näiden 10 framen aikana objektia niin, että framessa 5 objekti on sijainnissa 0.5,0. Tässä projektissa käytettiin kyseistä menetelmää luomaan sprite-animaatiot hahmoille. Animaattoria voi myös käyttää arvojen muuttamiseen tietyllä animaatiohetkellä, jolla voidaan esimerkiksi käskyttää äänijärjestelmää.

4 UNITY CANVAS

Canvas-järjestelmä julkaistiin Unity 3D:n versiossa 4.6. Canvas-järjestelmä kehitettiin helpottamaan valikkojen, heijastusnäyttöjen ja muiden näytöllä olevien elementtien näyttämistä. Canvas käyttää Rect Transform -komponenttia objektien sijaintien määrittämiseen näytöllä.

Canvaksella on kolme eri piirto asetusta. Screen Space - Overlay asettaa canvaksen suoraan näytölle, välittämättä muusta. Näytön koon muuttuessa osaa canvas muuttaa kokonsa automaattisesti. Screen Space - Camera liittää canvaksen tietyn matkan päähän kamerasta, johon se on liitetty, jolloin canvas piirretään aina kun kyseistä kameraa käytetään. Screen Space - Camera mahdollistaa perspektiivisten valikkojen ja heijastusnäyttöjen tekemisen. World Space mahdollistaa canvas-elementtien luonnin Unity 3D:n kolmiulotteiseen ympäristöön. Tällä voidaan piirtää canvas-objekteja eri paikkoihin pelimaailmassa. Kaikki canvaksen komponentit ovat mahdollisia animoida Unity 3D:n animaattorilla, joka mahdollistaa erilaisten efektien luonnin esimerkiksi valikkoon.

4.1 Canvaksen komponentit

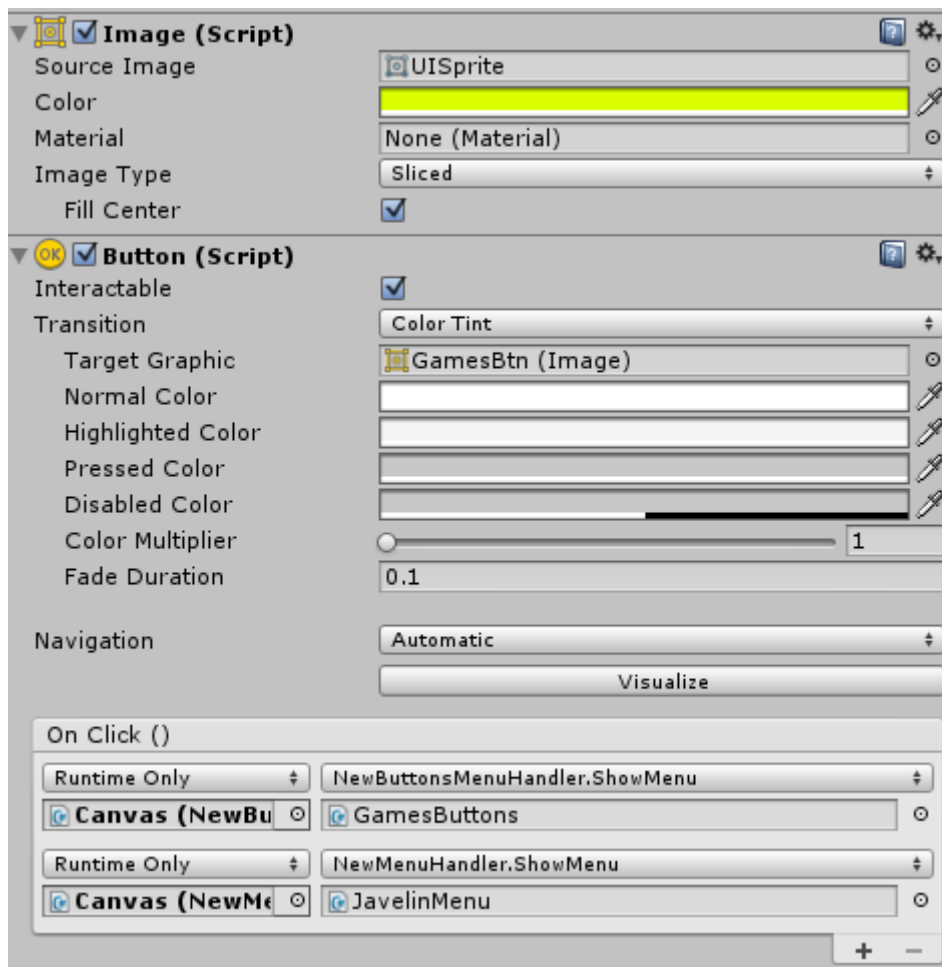
Canvaksen jokainen eri komponentti sisältää Rect Transform -komponentin, jolla määritetään, missä ja miten komponentti piirretään. Rect Transform sisältää eri ominaisuuksia, muun muassa ankkuroinnin, pivot-pisteen, rotaation, sijainnin sekä skaalauksen.

Paneeli

Paneeli eli panel-komponentti on suorakulmainen alue, johon voidaan lisätä eri canvas-komponentteja. Paneeli helpottaa objektien piirtämistä erikokoisille näytöille sen skaalautuvuuden takia. Paneeli voidaan asettaa läpinäkyväksi, jolloin siitä ei ole häiriötä näkyvyydelle. Paneelin voidaan liittää jokin myöhemmin mainittavista layout-komponenteista, jolloin objektit paneelin sisällä asettuvat oikein.

Painike

Painike eli button-komponentti on yksi tärkeimmistä canvas-komponenteista. Painike koostuu itse painike-luokasta ja sen sisällä olevasta teksti-komponentista. Kun painike tehdään, Unity 3D luo tapahtuma-komponentin, joka hallitsee painikkeiden tapahtumat, kun niitä painetaan. Painikkeen tapahtumaan on helppo liittää funktio, joka ajetaan painiketta painaessa. Funktio voi sijaita missä tahansa pelikomponentissa, kuhan se on samassa skenessä ja on julkisesti saatavilla. (Kuva 1.)



KUVA 1. Painike-komponentin tapahtumat ja ominaisuudet

Teksti

Teksti-komponentti eli text-komponentti on myös yksi tärkeistä komponenteista canvas-järjestelmässä. Teksti-komponentti mahdollistaa tekstin näyttämisen missä tahansa canvas-alueella.

Kuva

Kuva eli image-komponentti mahdollistaa tavallisten kuvien näyttämisen canvas-alueella. Image-komponentti, kuten muutkin komponentit, on mahdollista lisätä myös muiden komponenttien sisälle.

Liukuvalitsin

Liukuvalitsimella eli slider-komponentilla voidaan luoda rullautuvia alueita tai valintoja. Liukuvalitsin on hyvä, jos on paljon tekstiä eikä näytön tila riitä sen kaiken näyttämiseen kerralla, jolloin liukuvalitsimella voidaan rullata sisältöä. Toinen hyvä käyttötarkoitus liukuvalitsimelle on äänenvoimakkuuden säätö.

Valintapainike

Valintapainike eli Toggle-komponentti on valintaruutu, johon voidaan merkitä kyllä- tai ei-arvo. Valintapainiketta voidaan käyttää asetuksissa, äänet päällä tai ei. Oletustapahtuma tälle komponentille on `OnValueChanged(Boolean)`-funktio. Kun komponentti liitetään canvakseen, vaatii se toimiakseen boolean-tyyppisen muuttujan.

Syöttökenttä

Syöttökenttä eli `Inputfield` on tavallinen tekstikenttä, johon käyttäjä voi syöttää tekstiä ja numeroita. Oletustapahtumat tälle komponentille ovat `OnValueChange(string)` ja `EndEdit(string)`. `OnValueChange`-tapahtuma ajetaan jokainen kerta, kun kentän arvo on muuttunut. `EndEdit` ajetaan, kun komponentista siirrytään pois.

Tapahtumajärjestelmä

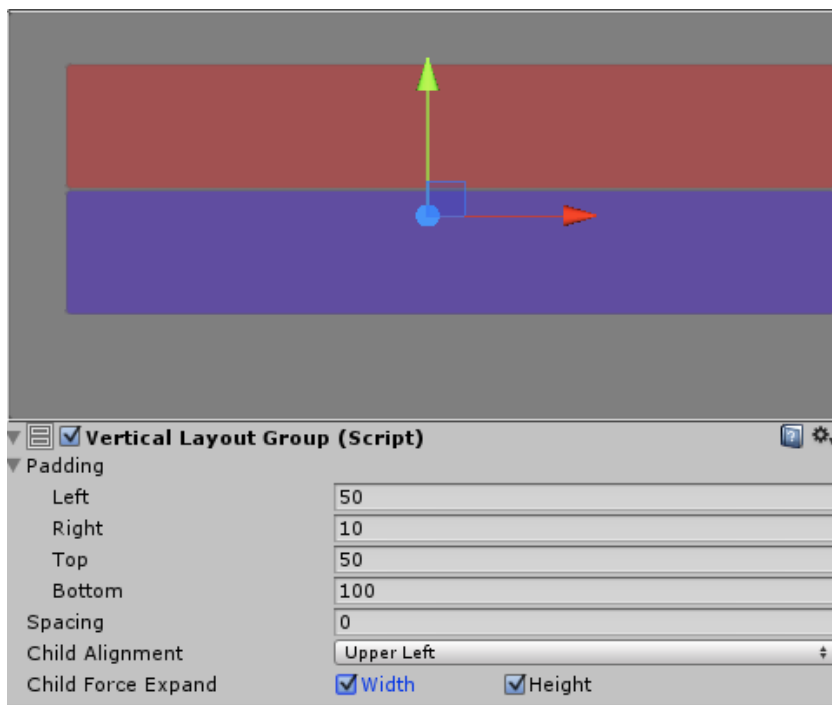
Tapahtumajärjestelmä eli `EventSystem` on canvas-komponentin tapahtuman hallintaan tarkoitettu komponentti. `EventSystem` hallitsee kaikkien komponenttien tapahtumat ja välittää ne eteenpäin tapahtumille merkittyihin komponentteihin.

4.2 Ulkoasun asetukset

Canvas mahdollistaa hyvin skaalautuvan käyttöliittymän luomisen ja sisältää useita eri komponentteja muiden komponenttien asetteluun. Jokainen layout group sisältää erilaisia parametreja, joilla voidaan määrittää lapsikomponenttien sijainti ja ulkoasu. Täytearvo eli padding mahdollistaa täyteen lisäämisen mihin tahansa laitaan layout groupissa. Väliarvo eli spacing-arvolla on mahdollista määrittää lapsikomponenttien väli.

Vertical Layout Group

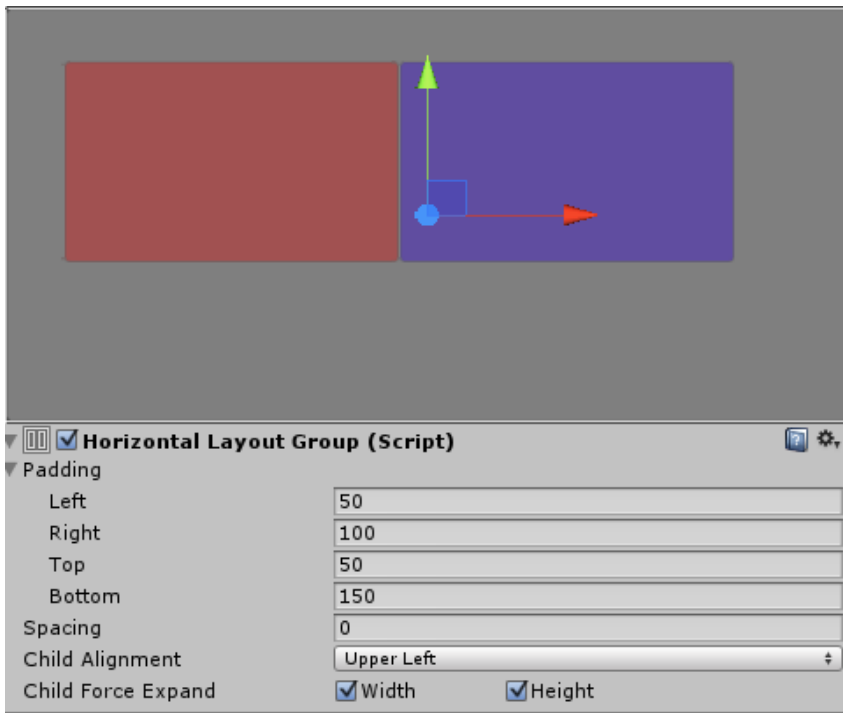
Vertical Layout Group asettaa kaikki sen lapsikomponentit pystysuuntaan linjaan (kuva 2).



KUVA 2. Vertical Layout Group

Horizontal Layout Group

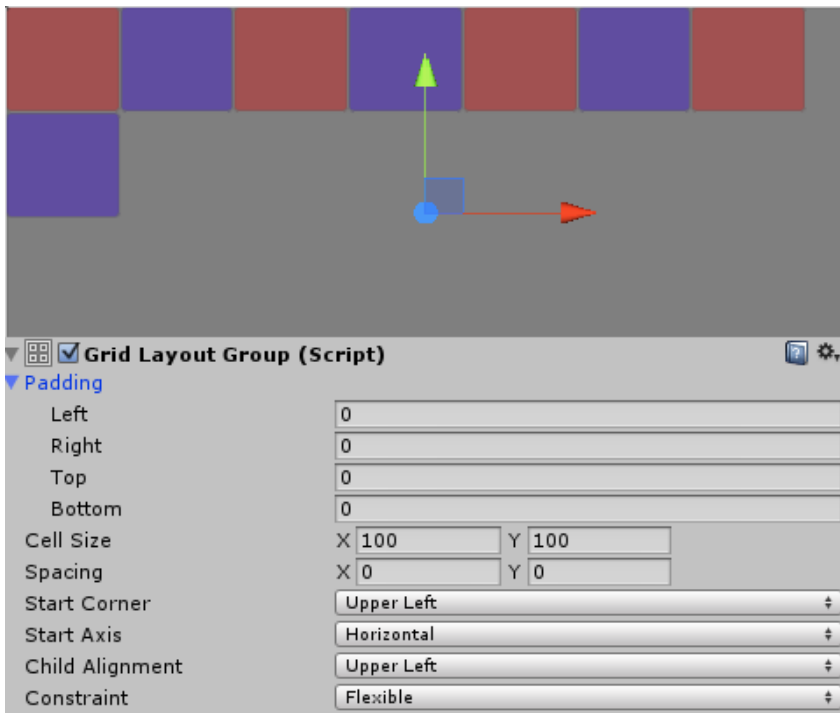
Horizontal Layout Group on ominaisuuksiltaan samanlainen kuin Vertical Layout Group, mutta Horizontal Layout Group asettaa lapsikomponentit rinnakkain (kuva 3).



KUVA 1. Horizontal Layout Group

Grid Layout Group

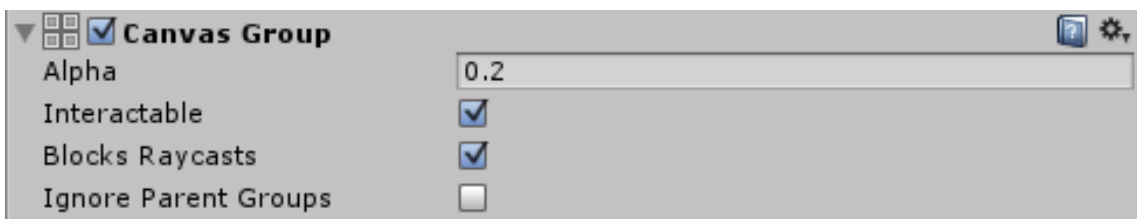
Grid Layout Group asettaa lapsikomponentit ruudukkoon. Grid Layout Group sisältää aiemmista layout-komponenteista tutut padding-ominaisuudet sekä asetukset ruudun aloituskulmaan ja suuntaan (kuva 4).



KUVA 2. Grid Layout Group

Canvas Group

Canvas Group on yksi tärkeimmistä ulkoasuun liittyvistä komponenteista. Näyttävää käyttöliittymää luodessa Canvas Group -komponentti mahdollistaa läpinäkyvyyden tason muuttamisen ja lapsikomponenttien käytön estämisen. Komponentin avulla voidaan luoda ponnahdusikkuna, jolloin taustalla olevia komponentteja ei pysty käyttämään, ennen kuin ikkuna on suljettu. (Kuva 5.)



KUVA 3. Canvas Groupin asetukset

5 GOOGLE ADWORDS

Google AdWords on Googlen kehittämä mainosjärjestelmä verkkosivujen ja eri sovellusten käyttöön. AdWords toimii pay-per-click-menetelmällä, eli kun käyttäjä painaa mainosta, saa sovelluksen tai sivun kehittäjä pienen summan rahaa. AdWords julkaistiin ensimmäisen kerran vuonna 2000, josta se on kehittynyt hyvin suosituksi mainosrajapinnaksi. Vuonna 2012 Google mainosten liikevaihto oli noin 43,7 miljardia USA:n dollaria. AdWords käyttää hyväkseen käyttäjistä tallennettuja tietoja muun muassa hakutuloksia, kieltä ja laitteen sijaintia. (7.)

Google AdWordsin käyttö Unity 3D:ssä on helppoa, sillä Unity 3D:hen on saatavilla avoimen lähdekoodin paketti nimeltä GoogleAds-mobile-plugins. Esimerkiksi koko ruudun mainoksen saa luoda helposti alla olevalla kutsulla. (6.)

```
using GoogleMobileAds.Api;
...
// Initialize an InterstitialAd.
InterstitialAd interstitial = new InterstitialAd("MY_AD_UNIT_ID");
// Create an empty ad request.
AdRequest request = new AdRequest.Builder().Build();
// Load the interstitial with the request.
interstitial.LoadAd(request);
```

Projektiin luotiin erillinen mainosten hallinta skripti eli AdHandler, joka hoitaa koko sovelluksen mainosten näyttämisen. AdHandler tarkkailee pelin tilaa ja tuo mainokset näkyviin sopivalla hetkellä, että mainokset eivät häiritsisi pelikokemusta. Koko ruudun mainos tulee näkyviin vain skenen vaihtuessa noin minuutin välein. Banner-mainos tulee näkyviin sovelluksen alkaessa ja on näkyvissä 35 sekuntia. Kun banner on piilotettu, lataa AdHandler automaattisesti uuden mainoksen ja 25 sekunnin kuluttua näytetään uusi mainos taas 35 sekuntia. Lataamalla mainokset taustalla ennen näyttöä pienennetään latausaikoja huomattavasti.

6 GOOGLE PLAY SERVICE

Google esitteli ensimmäisen kerran Google Play Servicen vuonna 2012. Google Play Service on taustapalvelun rajapinta Android-laitteille. Ensimmäisessä versiossa ei ollut kuin yksinkertainen yhteys Google+-rajapintaan. Myöhemmin rajapintaan tuli paljon uusia palveluita, muun muassa sijainti, kartat, Google Drive, Google Cast ja Google Wallet. (4.)

6.1 Google Play Service Unity 3D:ssä

Unity 3D:hen on saatavilla GitHubissa avoimen lähdekoodin liitännäinen Google Play Servicen käyttöön. Google Play Service mahdollistaa peliin kirjautumisen Google-tunnuksilla sekä erilaisten saavutusten ja tuloslistojen näytön. Google Play Service alustetaan sovelluksessa ennen sen käyttöä. Tähän tarkoitukseen luotiin skripti nimeltä `GooglePlayServiceManager`, joka säilyy mukana koko sovelluksen käynnissäoloajan. Kun objektille annetaan käsky `DontDestroyOnLoad(this)`, merkkää se objektin säilytettäväksi jokaisen skenen välillä. Jos sama skene, jossa objekti on alun perin luotu, ladataan uudelleen, luo se toisen kappaleen `GoolePlayServiceManagerista`. Tähän on saatavilla yksinkertainen koodi, joka varmistaa, että objektista ei luoda useampaa versiota kuin yksi. (5.)

```
if(FindObjectsOfType(GetType()).Length > 1) {  
    Destroy(gameObject);  
}
```

Kun objekti luodaan ja jos sovellus löytää jo olemassa olevan samanlaisen objektin, se tuhoaa itsensä. Näin kahta versiota samasta objektista ei synny. `GoolePlayServiceManager` sisältää useita muita funktioita Google Play Servicen yleisesti käytetyille toiminnoilla.

6.2 Saavutukset

Google Play Service sisältää pelejä varten luodun saavutusjärjestelmän. Jos pelaaja saavuttaa jonkun tietyn pisteen, voidaan hänelle myöntää saavutus. Saavutukset luodaan Google Play Servicen web-käyttöliittymässä. Saavutukselle saa sivulta tietyn sarjanumeron, jolla voidaan sovelluksessa saavutus merkata ansaituksi.

```
Social.ReportProgress("Koodi saavutukselle", 100.0f, (bool success) => {  
    // handle success or failure  
});
```

6.3 Tulostaulut

Google Play Service sisältää myös tulostaulut eli leaderboardit. Tulostaulut ovat tämän kaltaisissa peleissä hyvin tärkeitä, jos halutaan luoda kilpailua pelaajien välille. Kilpailu lisää mielenkiintoa peeliin. Tulostaulujen käyttöönotto on kohtalaisen helppo prosessi. Ensin luodaan Google Play Servicen uusi tulostaulu, tämän jälkeen sovelluksessa ajetaan tietty funktio kutsu, joka lisää tuloksen tulostauluun.

```
Social.ReportScore(value,"Taulun koodi", (bool success) => {  
    if(success) // Tuloksen lisäys onnistui  
    else { // Tuloksen lisäys epäonnistui  
    }  
});
```

Käyttäjälle tulostaulu voidaan näyttää yksinkertaisesti kutsumalla ShowLeaderBoardUI-funktiota. Funktioon voidaan syöttää parametrina tietyn tulostaulun koodi, jolloin funktio näyttää sen kyseisen tulostaulun.

7 PELIN SUUNNITTELU

Pelin tavoitteena oli luoda hauska, nopeutta vaativa urheilupeli, joka perustuu Nintendon Track and Field -peliin. Pelissä on useita erilaisia nopeutta ja tarkkuutta vaativia lajeja. Työskentely alkoi hyvissä ajoin ja eräs aiemmin tekemäni projekti sisälsi jo joitain ominaisuuksia, joita tähänkin tuli, joten käytin niitä hyväksi. Mekaniikka pelissä käyttää paljon hyväkseen Unity 3D:n fysiikkamoottoria ja suurin osa toiminnoista luotiin tämän ympärille.

Pelin kehitys tapahtui yksin, joten työtä oli paljon, mutta selkeä kuva pelin ominaisuuksista oli jo mielessä. Kehittäminen yksin on hieman helpompaa, kun tietää itse, missä mennään ja miten ominaisuudet on ajateltu toimivaksi. Tämän mittaluokan projektissa yksin kehittäminen onkin looginen vaihtoehto.

8 PELIN TOTEUTUS

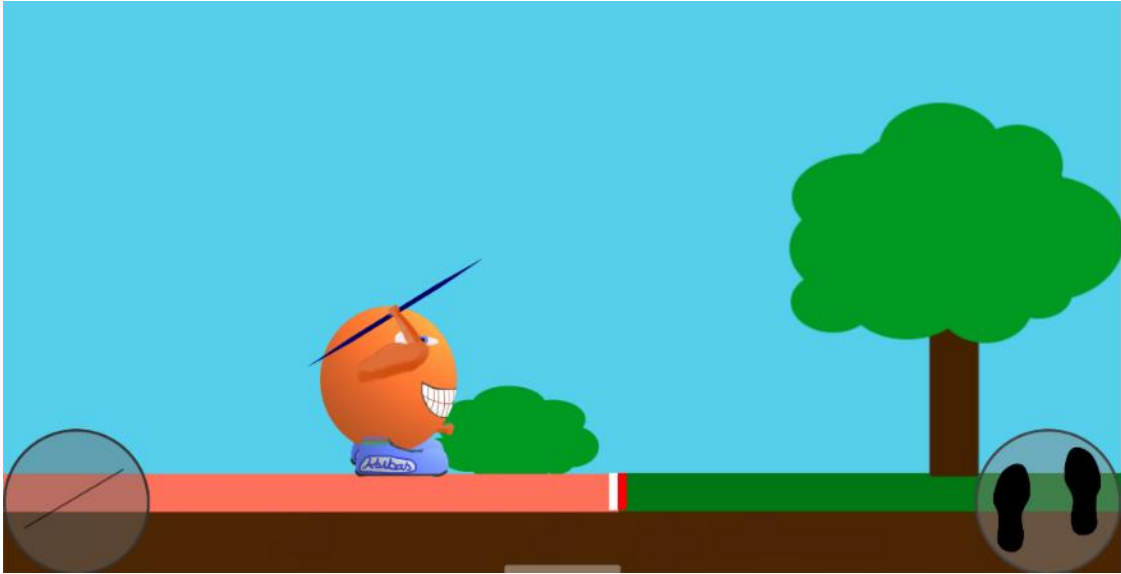
Kun idea oli valmiina, alkoi projektin toteutus. Ensimmäisenä aseteltiin kehityskenttä kuntoon, jossa voitaisiin testata eri komponentteja ja kehitettiin juoksu, keihäänheitto yms. skriptit.

8.1 Juoksu

Juoksu-skriptin luominen ei sinänsä ollut kovin vaikea, vaan eniten aikaa meni oikean tasapainon löytämiseen nopeuden lisäämisen ja hidastamisen välillä. Tasapainon löytäminen oli tärkeää, jotta pelaaja ei pystyisi saavuttamaan liian suuria nopeuksia helposti. Skripti kehitettiin erillisenä muista komponenteista. Näin sitä pystyy käyttämään suoraan muissakin lajeissa ilman muutoksia. Perusideana on, että joka kerta kun painiketta painetaan, lisätään tietyn verran vauhtia lisää. Jokaisella update-kierroksella nopeudesta vähennettiin tietty osa. Vauhdin vähennys tapahtui tietyllä prosentilla ja muutamalla muulla parametrilla. Näin pidettiin nopeus kurissa ja nopeuden kasvattaminen on paljon vaikeampaa.

8.2 Keihäänheitto

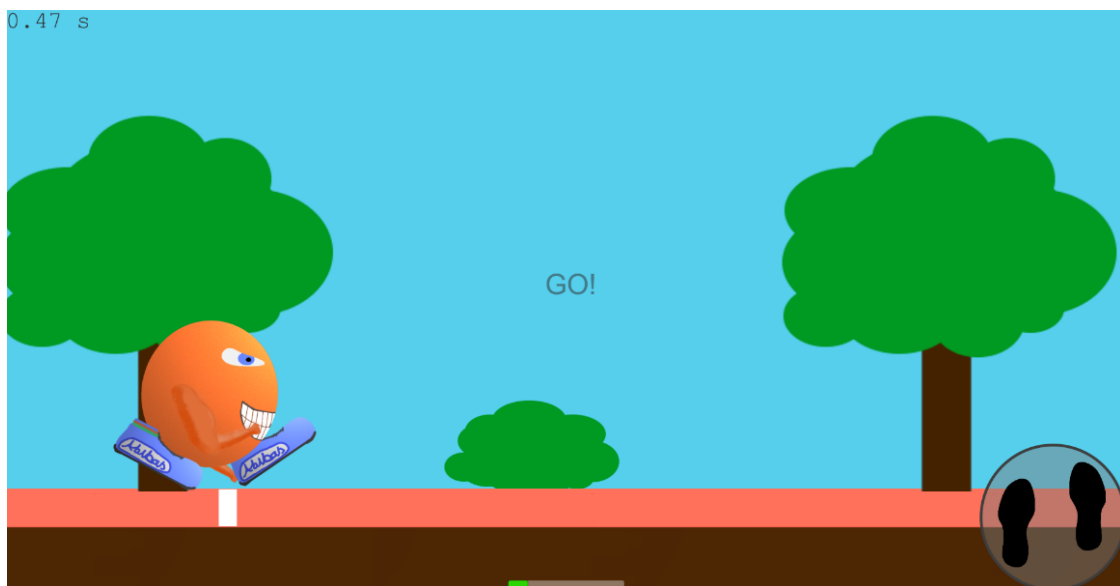
Keihäänheitossa oli ideana heittää keihästä mahdollisimman pitkälle. Keihään nopeus on suoraan yhteydessä juoksunopeuteen sekä heittokulmaan. Juoksijan nopeus muutettiin tietyn parametrein voimaksi, joka lisättiin keihääseen. Pelaajan täytyy säätää kulma heittäessä, jolloin oikea kulma on myös tärkeä, jotta keihäs lentää mahdollisimman pitkälle. (Kuva 6.)



KUVA 4. Keihäänheitto

8.3 100 m:n juoksu

Lajin idea on yksinkertaisempi kuin keihäänheiton. 100 m:n juoksussa ei tarvitse kuin nopeaa reaktiokykyä aloittaa juokseminen ja naputtaa juoksu-painiketta mahdollisimman nopeasti. Pelin alussa näytetään indikaattorilla 3, 2, 1, GO, milloin pelaaja saa lähteä juoksemaan. Liian aikaisesta lähdöstä tulee varaslähtö ja peli alkaa alusta. (Kuva 7.)

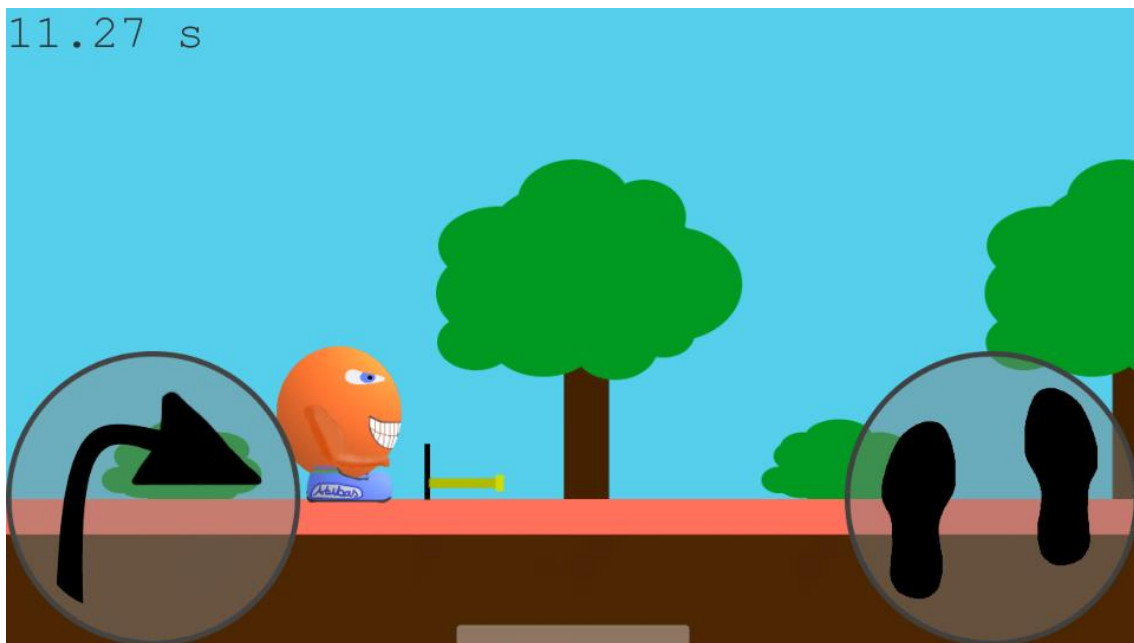


KUVA 5. 100 m:n juoksu

Kun pelaaja saapuu maaliin, näytetään tulos sekunteina. Valikosta on mahdollista katsoa muiden tuloksia tai kokeilla uudelleen.

8.4 110 m:n aitajuoksu

Lajin ideana on juosta 110 m mahdollisimman nopeasti, samalla hyppien aitojen ylitse. Esteeseen osuessa pelaajan vauhti hidastuu huomattavasti, mikä taas huonontaa aikaa. Pelin juoksu käyttää samaa skriptiä kuin muut aiemmat lajit. Hyppymekaniikka on luotu käyttämällä rigidbody-komponentin addForce-funktiota. (Kuva 8.)

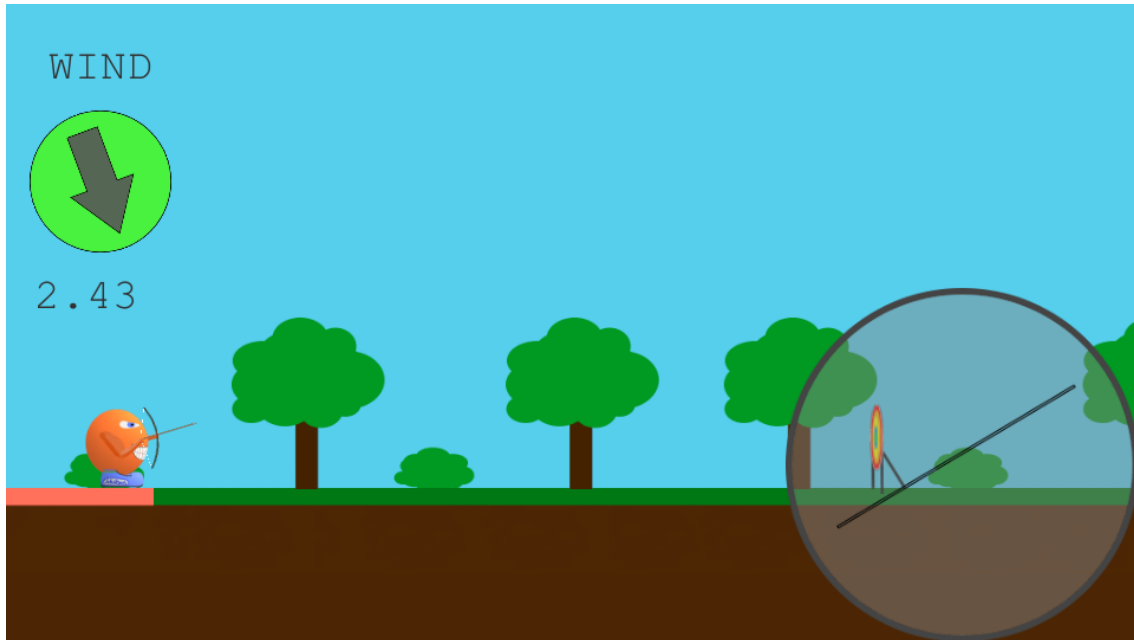


KUVA 6. 110 m:n aidat mobiililaitteella

8.5 Jousiammunta

Jousiammunta on hieman erilainen kuin muut lajit. Pelissä ei juosta ollenkaan, joten juoksu-skriptille ei ole tarvetta. Pelaajalla on kolme nuolta, jotka ammutaan tauluun. Jotta peli ei olisi liian helppo, luotiin siihen myös tuuli, joka vaikuttaa nuolen lentorataan. Tuulen suunta ja voima muuttuu jokaisella pelikerralla erilaiseksi, jotta pelaajan ei olisi helppo vain opetella tiettyä voimaa ja kulmaa, jolla osuu aina tauluun.

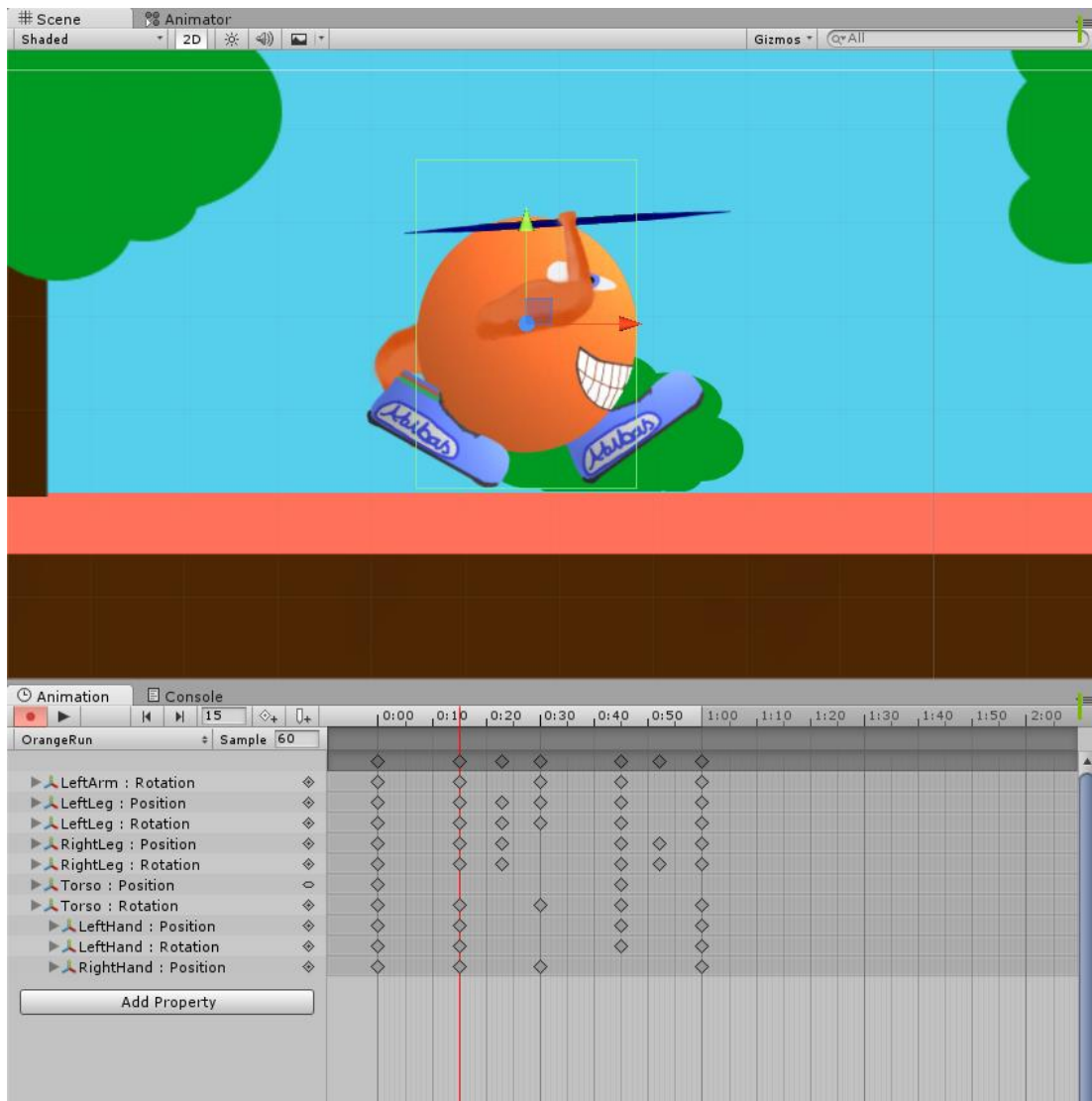
Nuolen kulma liikkuu tietyllä alueella ja pelaajan tulisi löytää paras kulma. Kun pelaaja painaa ammunta-painiketta, kulma lukittuu ja alkaa hahmo virittämään jouta (kuva 9). Mitä kauemmin painiketta pitää pohjassa, sen enemmän siinä on voimaa. Kun pelaaja on ampunut kaikki nuolet, näytetään tulos.



KUVA 7. Jousiammunta mobiililaitteella

8.6 Hahmon animointi

Hahmon animointiin käytettiin Unity 3D:n omaa animaattoria. Animaattorilla pystyy helposti luomaan animaatioita, varsinkin 2D-hahmoille. Hahmo luotiin irrallisilla raajoilla, joten animoimalla jokainen raaja erikseen saatiin hyvin ja monipuolisesti liikkuva hahmo. Esimerkkikuvassa näkyy, kuinka avainframe on määritetty kohdassa 0:10 tiettyyn kohtaan. (Kuva 10.)



KUVA 8. Hahmon animointi

8.7 Päävalikko

Päävalikosta oli tarkoitus luoda hyvin yksinkertainen, mutta samalla informatiivinen. Ensimmäiset versiot valikosta olivatkin väliaikaisia ratkaisuja skenen vaihdoksen tapahtumiseen.

Versio 1

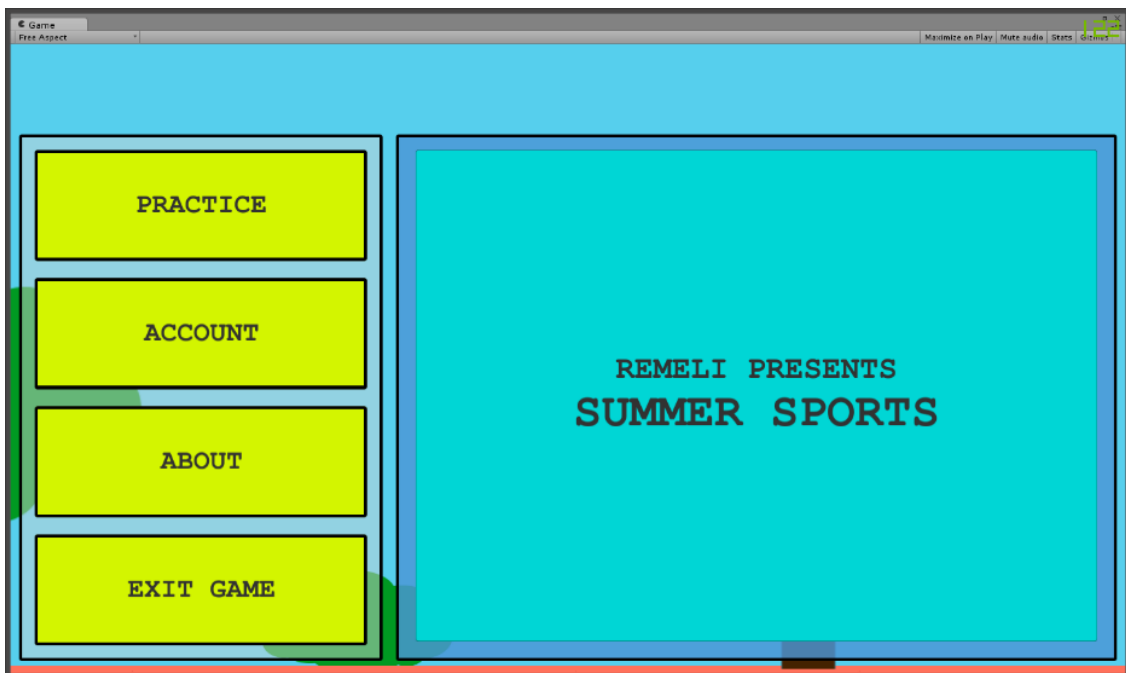
Versio 1:ssä liikuttiin sivuttaissuunnassa kameraa liikuttamalla. Taustaksi luotiin urheilukentän alue, jossa juoksumatolla näkyi haluttu laji. Valikko toimi kohtuullisen hyvin ja oli helppo käyttää, mutta ei aivan ollut tavoitteen mukainen. Kameraa liikuttava valikko ei myöskään ole kovin käytännöllinen.

Versio 2

Versio 2 luotiin käyttämällä uutta Canvas-järjestelmää sekä animaatioita valikkojen siirtymiin. Valikosta tuli visuaalisesti huomattavasti parempi kuin versio 1. Valikon toiminta oli osittain samantapainen kuin aiempi versio. Valikossa liikuttiin vetämällä sormeja näytön yli haluttuun suuntaan, jolloin näkymä vaihtui. Myöhemmin huomasin, että eleillä toimiva valikko ei ole kovin helppokäyttöinen, josta päästäänkin versioon 3.

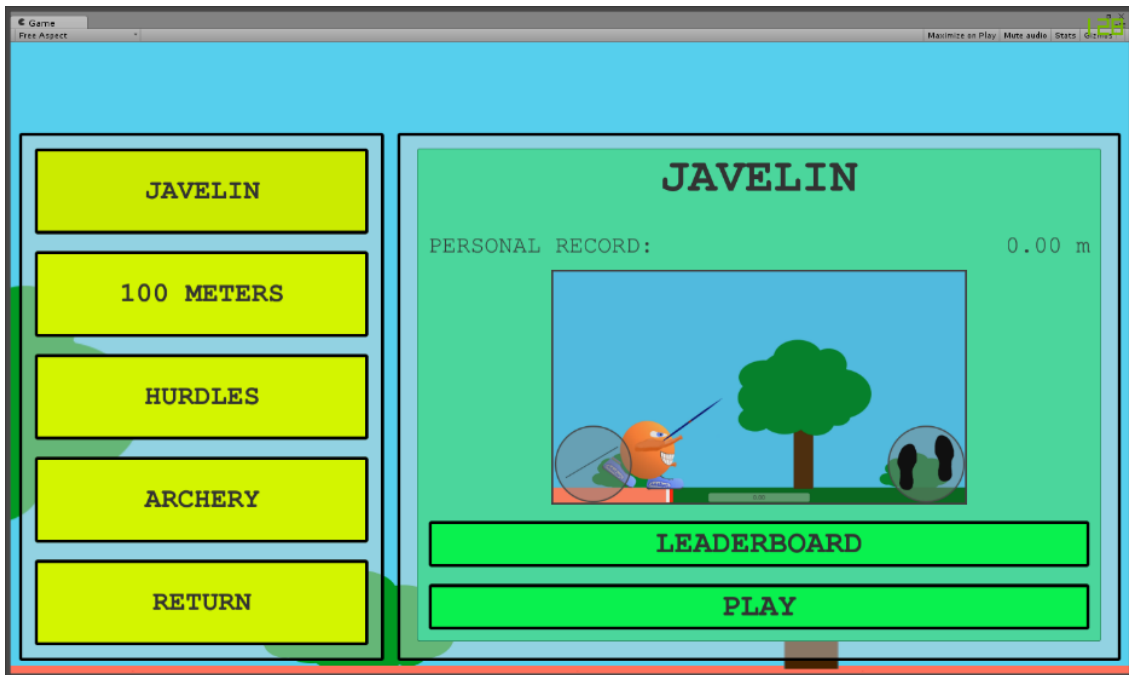
Versio 3

Versio 3 ei käyttänyt enää eleohjausta, vaan oli hieman perinteisempi painikkeilla toimiva valikko. Ruutu on jaettu kahteen eri osaan. Vasemmalla on painikevalikko, joka kattaa noin 30 % näytöstä, ja oikealla sisältövalikko, joka kattaa noin 70 % näytöstä. Painikkeet vasemmalla vaihtuvat valitun sisältöosion mukaan (kuva 11).



KUVA 9. Valikon aloitusnäky

Kun painikkeista valitaan pelit, tulee vasempaan laitaan näkyviin eri lajien painikkeet sekä paluu-painike. Oikealla puolelle tulee näkyviin valitun pelin infoalue, joka sisältää kuvankaappauksen pelistä, aloituspainikkeen, henkilökohtaisen ennätyksen ja leaderboard-painikkeen. (Kuva 12.)



KUVA 10. Pelien perusnäky

9 YHTEENVETO

Opinnäytetyön aiheena oli luoda Track and Field -tyylinen mobiilipeli. Projekti onnistui mielestäni hyvin ja oli joiltain osin haastava, mutta lopputuloksena syntyi kuitenkin hyvin pelattavissa oleva mobiilipeli. Projekti ei ole vielä kuitenkaan täysin valmis, mutta toivottavasti jossain vaiheessa peli on siinä kunnossa, että sen voi laittaa julkiseen jakeluun Google Play -kauppaan. Yhden miehen projektina tämä vaatii vielä paljon kehittämistä, testaamista ja grafiikan päivytystä. Vaikeudet projektissa oli saada oikeanlainen tasapaino hahmojen nopeuteen, voimaan ja tarkkuuteen. Graafisesti projekti oli haastava ja kehityksen aikana nähtiinkin mitä ihmeellisempiä hahmoja. Valikko rakennettiin muistuttamaan 80–90-luvun grafiikkaa, pelin retrohenkisyiden vuoksi. Valmiissa versiossa luultavasti tulemme näkemään kuitenkin huomattavasti parempaa, omalaatuisempaa ja erilaisempaa grafiikkaa kuin tässä työssä nähtiin.

LÄHTEET

1. Flappy Bird. 2015. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Flappy_Bird. Hakupäivä 2.12.2014.
2. Android. 2015. Wikipedia. Saatavissa: <http://fi.wikipedia.org/wiki/Android>. Hakupäivä 2.12.2014.
3. Unity 3D. 2015. Wikipedia. Saatavissa: http://fi.wikipedia.org/wiki/Unity_3D. Hakupäivä 26.3.2015.
4. Google Play Services. 2015. Wikipedia. Saatavissa: http://en.wikipedia.org/wiki/Google_Play_Services. Hakupäivä 31.3.2015.
5. Play-Games-Plugin-For-Unity. 2015. GitHub. Saatavissa: <https://github.com/playgameservices/play-games-plugin-for-unity>. Hakupäivä 15.12.2014.
6. Google-Mobile-Ads-SDK. 2015. GitHub. Saatavissa: <https://github.com/googleads/googleads-mobile-plugins>. Hakupäivä 20.1.2015.
7. AdWords 2015. Wikipedia. Saatavissa: <http://en.wikipedia.org/wiki/AdWords>.