



# **FLURRY ANALYTICS PELIKEHILYKSEN APUNA**

Rami Kuusisto

Opinnäytetyö  
Huhtikuu 2015  
Tietojenkäsittely  
Ohjelmistotuotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittely  
Ohjelmistotuotanto

KUUSISTO, RAMI:  
Flurry Analytics pelikehityksen apuna

Opinnäytetyö 28 sivua  
Toukokuu 2015

---

Flurry Analytics on Yahoo Mobile Developer Suiten osa, joka keskittyy analytiikkaan. Opinnäytetyössä kerrotaan Flurry Analytics SDK:n implementoimisesta sovellukseen, Flurry Analyticsin tarjoaman web-portaalin käytöstä, sekä siitä, miten näitä ominaisuuksia käytettiin toteutettaessa pelin Cabals: Legends analytiikkatoteutusta. Työssä tarkastellaan myös miten jo kehitettyä analytiikkatoteutusta voitaisiin käyttää pohjana vielä pidemmälle viedylle analytiikkatoteutukselle ja kuinka pystyttäisiin varautumaan mahdollisiin ongelmiin. Opinnäytetyön toimeksiantaja on Kyy Games Oy, joka on kehittänyt opinnäytetyössä tutkitun Cabals: Legends -pelin.

Flurry Analytics on monien mielestä mobiilisovellusten kehityksessä alan standardi mitä tulee analytiikkaan ja sillä on vuosien kuluessa kertynyt laaja käyttäjäkunta. Opinnäytetyössä esitellään Flurry Analyticsin soveltuvuutta pelikehityksen avuksi.

Opinnäytetyössä tarkastellaan Flurry Analyticsin hyvin kenttätestattua järjestelmä, jonka käyttöönotto on verrattain nopeaa ja melko helppoa. Tarkasteltavina ovat myös Flurry Analyticsin rajoitukset ja heikkoudet, joiden takia kehittäjä voi helposti siirtää huomionsa muualle etsiessään palveluntarjoajaa analytiikkatoteutukselle.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Business Information Systems  
Option of software development

KUUSISTO, RAMI:  
Using Flurry Analytics in Game Development

Bachelor's thesis 28 pages  
May 2015

---

This bachelor's thesis focuses on how to use Flurry Analytics in game development. Flurry Analytics is a subset of tools found in Yahoo Mobile Developer Suite that focuses on analytics. The thesis itself concentrates on the implementation of Flurry Analytics to an application, the use of Flurry Analytics' web-portal and how these tools were used while realizing analytics implementation for a game called Cabals: Legends. The thesis also considers how the analytics implementation realized here could be taken even further and how potential problems could be minimized. The thesis was commissioned by Kyy Games Ltd, the company behind the development of Cabals: Legends.

Given the quantity of developers using Flurry Analytics, it has proven to be a field-tested system and considered by many as the industry standard analytics provider in mobile development. This thesis takes a different angle on it, approaching it from a mobile game developer's perspective.

The results of the thesis confirm that while Flurry Analytics is a robust system, it has its weaknesses and limits that may force one to consider using some other analytics provider.

---

Key words: analytics, flurry, game development, events, parameters

## SISÄLLYS

1	JOHDANTO.....	6
2	FLURRY ANALYTICS.....	8
2.1	Yleistä .....	8
2.2	Projektin luominen Flurry Analytics-portaaliin.....	9
2.3	Käyttö sovelluksessa.....	9
2.3.1	Implementointi .....	9
2.3.2	Tapahtumien lisääminen .....	10
2.3.3	Muita tärkeimpiä ominaisuuksia .....	12
2.4	Flurry Analytics-projektin hierarkia .....	13
2.4.1	Dashboards .....	13
2.4.2	Usage.....	15
2.4.3	Retention .....	15
2.4.4	Audience .....	16
2.4.5	User Acquisition .....	17
2.4.6	Events.....	17
2.4.7	Errors.....	18
2.4.8	Technical .....	18
2.4.9	Manage.....	18
3	HAVAINNOT EDELLISESTÄ FLURRY ANALYTICS- INTEGRAATIOSTA .....	20
3.1	Yleistä .....	20
3.2	Mitä haettiin.....	21
3.3	Miten siinä onnistuttiin .....	22
3.4	Havaitut ongelmat.....	23
4	MITEN PARANTAA FLURRY ANALYTICS-INTEGRAATIOTA .....	24
4.1	Mitä halutaan tietää.....	24
4.2	Miten siihen päästään.....	25
4.3	Miten vältetään ongelmat.....	26
5	JOHTOPÄÄTÖKSET JA POHDINTA .....	27
	LÄHTEET .....	28

## ERITYISSANASTO

Flurry Analytics SDK	Flurry Analyticsin ohjelmistokehitystyökalut, saatavilla eri alustoille. SDK on lyhenne sanoista Software Development Kit.
AndroidManifest	XML-muotoinen tiedosto, jossa määritellään Android-sovelluksen vaatimat ominaisuudet. Pakollinen osa Android-sovellusta.
FlurryAgent	Flurry Analytics SDK:sta löytyvä luokka, joka pitää sisällään Flurry Analyticsiin liittyviä kutsumuksia.
HashMap	Kirjasto-tyyppinen tietorakenne Java-ohjelmointikielessä. Pitää sisällään listan avaimia ja niiden arvoja.
Spark Charts	Flurry Classic-dashboardista löytyvä tietolaa- tikko, josta näkee nopeasti määrätyn aikavälin sessiot, uudet käyttäjät, session mediaanipituu- den sekä keskimääräisen arvion aktiivisista käyttäjistä.
Churn point	Se hetki, kun peliä aktiivisesti pelannut pelaaja lopettaa pelaamisen lopullisesti.
MMO	Massiivinen monen pelaajan verkkopeli. Ly- henne sanoista Massively Multiplayer Online.

## 1 JOHDANTO

Pelien kehittäminen on harpponut suuria askeleita viime vuosina ja pelikehittäjien on tullut panostaa käyttäjätutkimukseen resursseja yhä suuremmissa määrin varmistaakseen kehittämiensä pelien menestyksen. Käyttäjätutkimus ja analytiikka eivät rajoitu ainoastaan pelin käytettävyyden sekä pelikokemuksen parantamiseen, vaan tärkeitä osa-alueita ovat myös esimerkiksi käyttäjien ostokäyttäytymisen, käyttäjien pelissä pitämisen, pelin latausaikojen sekä pelin leviämisen tarkastelu ja analysointi.

Opinnäytetyöni toimeksiantajana toimii vuonna 2009 perustettu tamperelainen Kyy Games Oy, joka kehittää ja julkaisee pelejä niin mobiili-, web- kuin tietokonealustoille. Yritys yhdistyi vuoden 2013 lopulla Dicework Gamesin kanssa ja työllistää tällä hetkellä noin 18 henkilöä. Yrityksen erityisosaamisena tunnetaan digitaalisten pelien lisäksi myös lauta- ja korttipelit. Tätä opinnäytetyötä varten tutkin Kyy Games Oy:n peliin Cabals: Legends sisällytettyä Flurry Analytics-implemентаatiota. Cabals: Legends julkaistiin Windows Phone 8-alustalle vuonna 2014.

Flurry on alun perin vuonna 2005 perustettu yritys, jonka Yahoo hankki haltuunsa vuonna 2014 (Flurry, 2014) liittääkseen Flurryn tarjoamat palvelut kehittämäänsä Yahoo Mobile Developer Suiteen. Flurryn tehtävänä on ollut omien sanojensa mukaan optimoida käyttäjän mobiilikäyttökokemus parempien sovellusten ja yksilöityjen mainosten avulla. (Flurry, 2014) Yrityksen tarjoamat palvelut ovat jaettu kolmeen osaan: analytiikka (Flurry Analytics), monetisaatio (Flurry for Publishers) sekä mainostus (Flurry for Advertisers).

Tämä opinnäytetyö keskittyy Flurryn palveluiden analytiikkaosuuteen, nimeltään Flurry Analytics, joka on alan standardi mobiilialustoilla ja jota käyttää yli 170 000 kehittäjää. (Flurry, 2015) Opinnäytetyössä käytetyt koodiesimerkit ovat Flurry Analyticsin Android-versioon liittyviä, mutta erot muiden alustojen Flurry Analytics-versioihin ovat varsin mitättömiä. Suurimmat erot ovat syntaksissa ja siinä, miten Flurry Analyticsin kirjastot ovat luotu ja kuinka ne tulee sisällyttää sovelluksen projektiin.

Opinnäytetyössäni tavoitteeni on tutkia ja selvittää kuinka Flurry Analyticsin käytöllä voidaan saada tarvittavia tuloksia pelin kehittämiseksi sen eri osa-alueilla. Kyy Games Oy:lle tavoitteena on saada parempi ymmärrys kerätystä tiedosta ja siitä, miten jatkossa voidaan kerätä tehokkaammin yhä hyödyllisempää ja tarkempaa tietoa. Henkilökohtai-

nen tavoitteeni on päästä aiempaa syvemmälle analytiikan maailmaan ja ymmärtää paremmin eri tapahtumien välinen yhteys. Tavoitteenani on myös hankkia parempi valmius toteuttaa aiempaa parempia analytiikkaimplementaatioita.

Opinnäytetyöni tarkoituksena on saada vielä parempi ymmärrys siitä, miten Flurry Analyticsin avulla voidaan tarkastella ja analysoida pelikehityksen kannalta tärkeitä elementtejä kvantitatiivisen tiedon kautta. Tarkoituksena on myös saada parempi kuva siitä, kuinka kvantitatiivisen tiedon keräämistä Flurry Analyticsin avulla voidaan parantaa.

## 2 FLURRY ANALYTICS

### 2.1 Yleistä

Datan keräämiseen ja sen analysointiin tarkoitettu osa, Flurry Analytics, on ilmainen ja hyvin skaalautuva. Sen yksinkertainen käyttö ei tarvitse suuria ponnistuksia. Esimerkiksi iOS:llä Flurry Analyticsin yksinkertaisin käyttö tarvitsee vain Flurry Analytics-kirjaston lisäämisen projektiin ja kolme koodiriviä. Jo tällä lisäyksellä Flurry kerää dataa käyttäjistä, laitteista, sessioista ja saadaan avuksi Flurry Persona -profiilit, joiden avulla pystytään päättämään millaiset ihmisryhmät käyttävät sovellusta. Nämä tiedot pohjautuvat Flurryn jo tuntemiin käyttäjiin muista mobiilisovelluksista sekä niiden pohjalta laadittuihin ennustuksiin.

Flurry Analyticsin palvelut eivät kuitenkaan jää tähän, vaan sillä voidaan seurata sovelluksesta laukaistuja tapahtumia, joihin voidaan myös antaa parametreja tarkemman tiedon saamiseksi. Tarkkaan harkittujen tapahtumien avulla saadaan parempi kuva käyttäjien käyttäytymisestä niin sovelluksen käytön kuin ostotapahtumienkin osalta. Tapahtumiin sisällytetyillä parametreilla voidaan vielä tarkentaa saatuja tietoja, esimerkiksi sovelluksessa tapahtuvaan ostotapahtumaan voidaan liittää parametriksi ostettu tuote, jolloin voidaan tarkastella kuinka usein erilaisia tuotteita ostetaan.

Tapahtumien avulla voidaan myös luoda suppiloita, joissa tarkastellaan kuinka moni käyttäjä kulkee määrätyn tapahtumaketjun läpi. Käytännössä tämä tapahtuu siten, että tarkastellaan kuinka usein tapahtumat tulevat tietyssä järjestyksessä. Pelikehityksessä tästä on hyötyä esimerkiksi siinä, että voidaan tarkastella missä kohdassa käyttäjän mielenkiinto peliä kohtaan laskee niin paljon, että käyttäjä lähtee pelistä eikä enää palaa takaisin. Tällöin osataan kiinnittää huomiota ongelmakohtaan ja tehdä siihen tarvittavia parannuksia käyttäjän mielenkiinnon ylläpitämiseksi.

Yhteneväinen ominaisuus lähes kaikille Flurry Analytics-projektin välilehtien ominaisuuksille on mahdollisuus luoda segmenttejä, joilla voidaan rajata käyttäjäkuntaa ja siltä kerättyä dataa halutulla aikavälillä. Tällöin on mahdollista tutkia miten erilaiset käyttäjät reagoivat tutkittaviin asioihin.

Lähes kaikki Flurry Analyticsiin lähetetty ja tallennettu tieto voidaan myös ladata CSV-muotoisena tiedostona Flurryn palvelimilta paikallista tarkastelua varten. Tämä mahdollistaa myös tietojen tallentamisen paikallisesti, mikäli siihen on tarvetta.

## **2.2 Projektin luominen Flurry Analytics-portaaliin**

Käyttäjän tulee ensin luoda tunnus Flurry Analyticsin hallinnointisivustolle (Flurry Analytics, 2015). Tämän jälkeen voidaan luoda sovellukselle uusi projekti, johon määritellään käyttöjärjestelmä jossa sovellus pyörii. Mikäli sovellus on kehitetty toimimaan useammalla alustalla, kannattaa jokaiselle tuetulle alustalle luoda oma projekti ja ohjata sovelluksesta kerätty data oikeaan paikkaan. Flurry Analytics ei tue useamman sovelluksen datan sijoittamista yhden Flurry Analytics-projektin alle, joten väärin käytettynä projektiin voi ilmetä rikkiäistä dataa. Alustakohtainen tietojen erittely tulisi tällöin myös todella haastavaksi, mikäli kaikkien sovelluksen versioiden tieto kerättäisiin saman Flurry Analytics-projektin alle.

Jokaisella luodulla projektilla on Application Key, joka on projektin uniikki 20-merkkinen merkkijono sisältäen kirjaimia ja numeroita. Sovelluksessa käytettävä Flurry Analytics alustetaan Application Key:tä käyttäen, jolloin sovelluksesta Flurryn palvelimille lähetetty data osataan ohjata oikeaan projektiin.

## **2.3 Käyttö sovelluksessa**

### **2.3.1 Implementointi**

Flurry Analyticsin implementoiminen mobiilisovellukseen on teknisesti melko helppoa. Flurryn tarjoamat käyttöohjeet ovat myös hyvin selkeät ja kattavat. Projektin luonnin jälkeen tulee ladata kyseiselle alustalle Flurry Analytics SDK, josta löytyvät tarvittavat kirjastot Flurry Analyticsin integroimiseen sovellukseen. Android-alustalle implementoitaessa tulee myös varmistaa, että AndroidManifestissa on sisällytetty tuki vähintään Internet-yhteydelle. Kun kirjastot ovat sisällytetty sovelluksen projektiin, kutsutaan Flurry Analyticsin alustamisfunktiota sovellusta käynnistettäessä käyttäen Flurry Analytics-projektin Application Key:tä, jonka avulla kerätty data saadaan talteen oikean pro-

jektin alle Flurryn palvelimilla. Androidilla Flurry Analyticsin yksinkertaisin implementaatio on kirjastojen sisällyttämisen jälkeen hyvin yksinkertainen. (Kuva 1)

```
//If you are shipping an app, extend the Application class
//if you are not already doing so:

public class MyApplication extends Application
{
    @Override
    public void onCreate()
    {
        super.onCreate();

        // configure Flurry
        FlurryAgent.setLogEnabled(false);

        // init Flurry
        FlurryAgent.init(this, MY_FLURRY_APIKEY);
    }
}
```

Kuva 1. Yksinkertaisin Flurry Analytics-integraatio Android-käyttöjärjestelmässä (Flurry, 2015, muokattu)

Flurry Analytics SDK:sta löytyvän FlurryAgentin metodilla setLogEnable määritellään halutaanko Flurry Analyticsin toiminnasta lokeja sovellusta suorittaessa. Toteutusta implementoitaessa ja testattaessa on suositeltavaa kutsua metodia true-parametrilla, jotta saadaan paras mahdollinen tieto missä asiat menevät vikaan, mikäli menevät. Kun toteutus on huolella testattu, voidaan parametri muuttaa falseksi.

### 2.3.2 Tapahtumien lisääminen

Tapahtumien kirjaaminen lähetettäväksi tapahtuu kutsumalla FlurryAgentin metodia logEvent. Tämä kirjaa määritellyn nimisen tapahtuman tapahtuneeksi ja valmiiksi lähetettäväksi. Yksinkertaisimmillaan tapahtuman lisääminen on yhden rivin kutsu, jossa kerrotaan ainoastaan tapahtuman nimi. (Kuva 2)

```
// Kutsutaan logEvent-funktiota tapahtuman kirjaamiseksi
// antaen sille tapahtuman nimen
FlurryAgent.logEvent("MyAwesomeEvent");
```

Kuva 2. Flurry Analytics-tapahtuma kutsuen pelkkää tapahtuman nimeä (Flurry, 2015, muokattu)

Mikäli tapahtumaan halutaan lisätä parametreja, tulee ne toteuttaa alustakohtaisesti määritellyillä tavoilla. Androidilla tämä kyseinen tapa on HashMap, johon lisätään halutut parametrit ja niiden arvot. Parametrien lisäämisen jälkeen HashMap syötetään mukaan kutsuun.

```
// Luodaan HashMap parametreille
Map<String,String> parameters = new Hashmap<String, String> ();

// Syötetään parametrit luotuun HashMappiin
parameters.put("MyFirstParameter", "Some Value");
parameters.put("MySecondParameter", "Some Other Value");

// Kutsutaan logEvent-funktiota tapahtuman kirjaamiseksi ja
// annetaan sille luotu HashMap, joka sisältää halutut parametrit
FlurryAgent.logEvent("MyAwesomeEventWithParameters", parameters);
```

Kuva 3. Flurry Analytics-tapahtuma käyttäen parametreja (Flurry, 2015, muokattu)

Tapahtumia voidaan myös antaa ajastettuna, jolloin Flurry Analytics-projektiin tulee myös tiedot siitä kuinka kauan kyseinen tapahtuma kesti alusta loppuun. Ajastettua tapahtumaa voidaan kutsua joko parametreilla tai ilman, aivan kuten normaalia tapahtumaa kutsuttaessa. Tapahtumaa kutsuttaessa tulee vain lisätä vielä yksi boolean parametri true:na kutsun loppuun. Kun ajastettu tapahtuma halutaan lopettaa, kutsutaan FlurryAgentin endTimedEventtiä ajastetun tapahtuman nimellä.

```
// Tapahtuma ilman parametreja

// Kutsutaan tapahtumaa nimellä ja annetaan sille true-arvoinen
// boolean ajastetun tapahtuman merkiksi
FlurryAgent.logEvent("MyTimedEvent", true);

// Lopetetaan ajastettu tapahtuma kutsumalla sitä nimeltä
FlurryAgent.endTimedEvent("MyTimedEvent");
```

Kuva 4. Flurry Analytics-tapahtuma ajastettuna (Flurry, 2015, muokattu)

```
// Tapahtuma parametrien kanssa

// Luodaan HashMap parametreille
Map<String,String> parameters = new Hashmap<String, String> ();

// Syötetään parametrit
parameters.put("MyFirstParameter", "Some Value");
parameters.put("MySecondParameter", "Some Other Value");

// Kutsutaan tapahtumaa nimellä, annetaan sille parametrit ja
// true-arvoinen boolean ajastetun tapahtuman merkiksi
FlurryAgent.logEvent("MyTimedParameterEvent, parameters, true");

// Lopetetaan ajastettu tapahtuma
FlurryAgent.endTimedEvent("MyTimedParameterEvent");
```

Kuva 5. Flurry Analytics-tapahtuma ajastettuna, parametrien kanssa (Flurry, 2015, muokattu)

### 2.3.3 Muita tärkeimpiä ominaisuuksia

Flurry Analytics sisältää myös muita hyödyllisiä ominaisuuksia jo aiemmin mainittujen lisäksi. Flurry Analyticsin avulla voidaan esimerkiksi tutkia kaupungin tarkkuudella mistä käyttäjät käyttävät sovellusta. Tämä ominaisuus vaatii Androidilla muutoksia AndroidManifestiin, jotta tarkempi GPS-paikannus olisi mahdollinen. Ilman tätä tar-

kempaa paikannusta Flurry Analytics käyttää käyttäjän IP-osoitetta määrittääkseen käyttäjän sijainnin maan tarkkuudella.

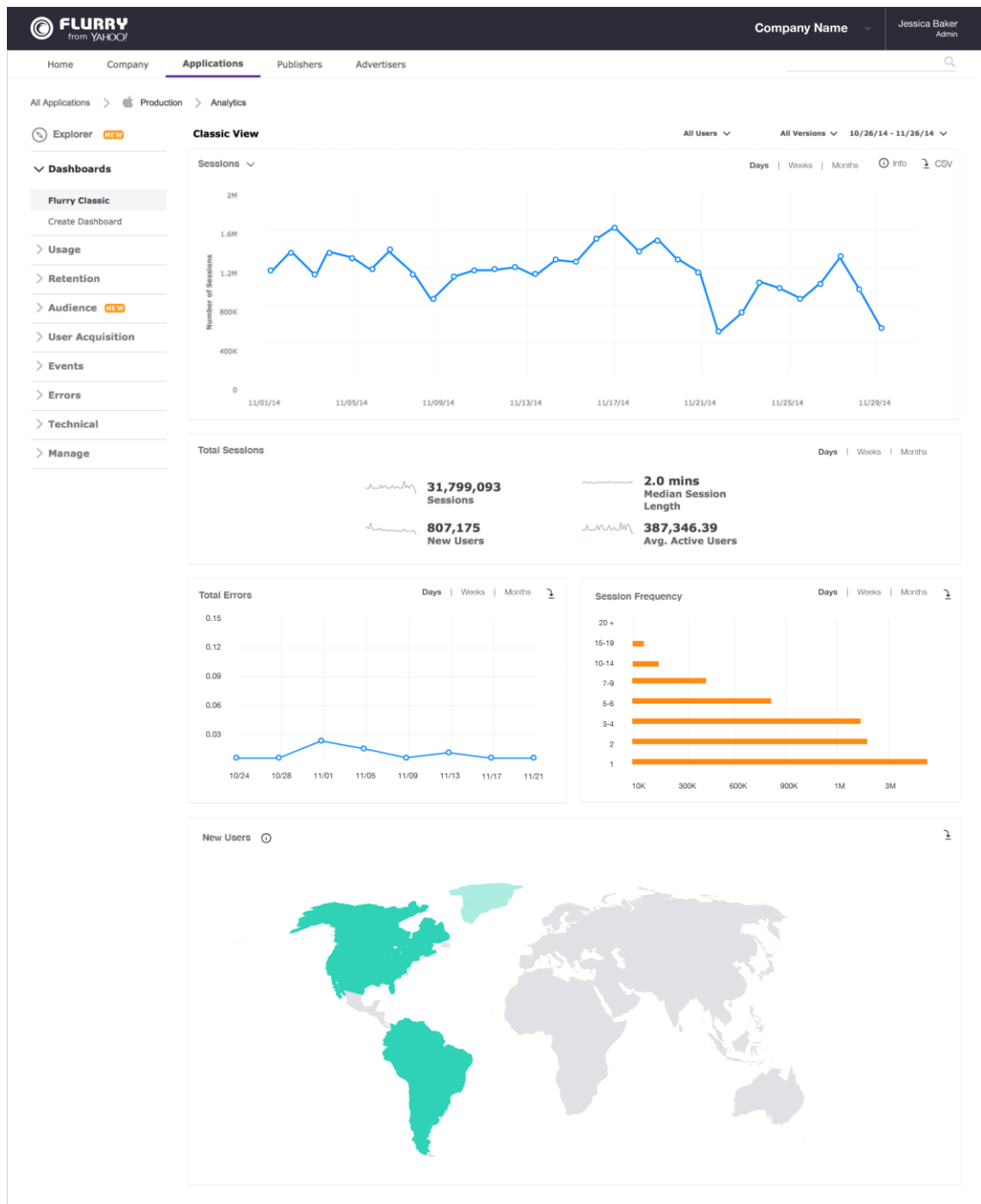
Mikäli tarkempi paikannus halutaan pois käytöstä, voidaan kutsua FlurryAgentin metodia `setReportLocation` parametrilla `false`. Halutessaan voi myös määrittää manuaalisesti käyttäjän sijainnin kutsumalla FlurryAgentin metodia `setLocation`, joka ottaa ensimmäisenä parametrina leveysasteen float-muodossa ja toisena parametrina pituusasteen niin ikään float-muodossa. (Flurry, 2015)

FlurryAgentin metodi `setUserID`, joka ottaa vastaan String-parametrin, on hyödyllinen, mikäli halutaan tarkempaa tietoa eri käyttäjien tapahtumista. Tämän avulla eri sessiot voidaan yhdistää yhteen käyttäjään ja saada parempi kuva tämän tekemisistä sovelluksessa eri aikoina. Parametriksi ei kuitenkaan saa laittaa tietoa mikä voidaan yhdistää tiettyyn henkilöön, kuten esimerkiksi Android ID, koska tämä rikkoo Flurryn käyttöehtoja. Flurry suositteleeikin anonymisoimaan lähetettävän tunnisteiden esimerkiksi MD5 tai SHA256 hashilla. (Flurry, 2015)

## **2.4 Flurry Analytics-projektin hierarkia**

### **2.4.1 Dashboards**

Avatessasi luomasi projektin Flurry Analyticsin kehittäjä sivulla, eteesi avautuu Flurry Classic-dashboard näkymä. (Kuva 6) Tähän näkymään kuuluvat sessioiden ja aktiivisten käyttäjien listaaminen, Spark Charts, sovelluksessa tapahtuneet virheet, sovelluksen käyttötiheys ja uusien käyttäjien sijainti kartalla.

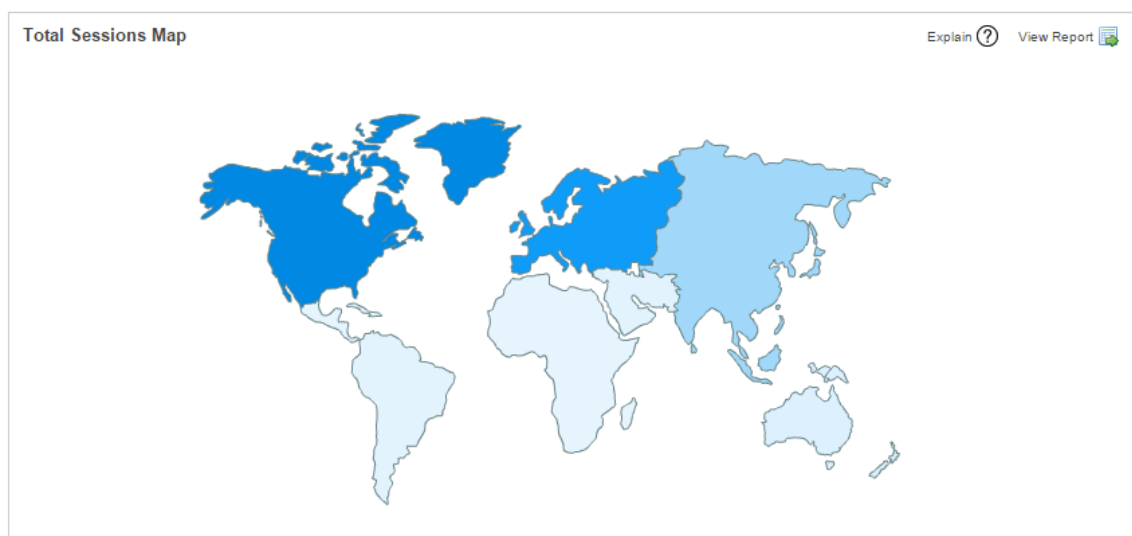


Kuva 6. Flurry Analytics Flurry Classic-dashboards (Flurry, 2015)

Dashboards-välilehdellä on myös mahdollista luoda omia dashboard-näkymiä. Flurry tarjoaa tähän tarkoitukseen useita valmiita pohjia, muun muassa pohja käyttäjien hankkimisen ja säilyttämisen tarkkailuun sekä erilaiset pohjat eri monetisaatiomalleihin perustuvien sovellusten tarkkailuun.

## 2.4.2 Usage

Välilehden Usage alta löytyy tarkempaa dataa sovelluksen käytöstä ja käyttäjistä. Uusien ja aktiivisten käyttäjien määrää voi seurata sekä viiva- että pylväskaaviona, joiden lisäksi nähtävillä on myös kartta käyttäjien sijainnista. Sessioiden määrästä löytyy vastaavat kaaviot kuin käyttäjien tarkastelusta. Sessioiden pituuksien tarkasteluun on käytettävissä pylväskaavio yleisimmistä pituuksista, aikajanelle sijoitetut kyseisen päivän mediaanin mukaiset pituudet, päivittäisten mediaanien tarkat pituudet ja koko päivänä sovelluksessa käytetty aika, sekä kartalla näytetyt maanos- ja maakohtaiset mediaanit. Käytön tiheyttä voidaan tarkastella päivittäisen, viikoittaisen ja kuukausittaisen ympyräkaavion avulla, sekä aikajanelle sijoitetun viivadiagrammin avulla. Saatavilla on myös tietoa sovelluksen versioista, jota tarkastelemalla voidaan tutkia eri versioiden määriä eri aikoina.



Kuva 7. Kuvankaappaus Usage-välilehden Sessions-osion karttanäkymästä (Kyy Games Oy, Flurry, 2015, muokattu)

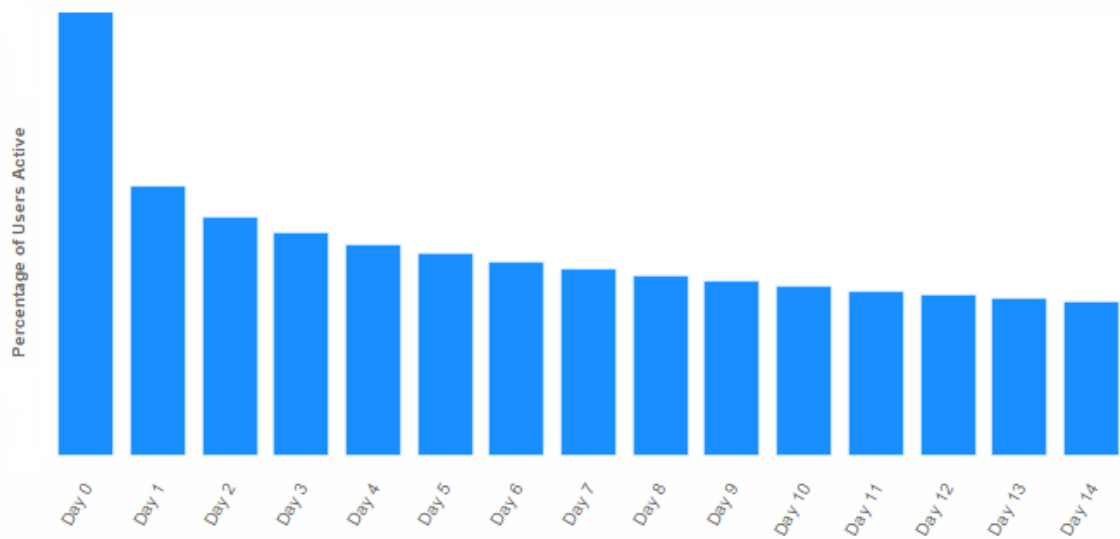
## 2.4.3 Retention

Käyttäjien pitämistä sovelluksessa voidaan seurata Retention-välilehden alta löytyvien työkalujen avulla. Return Rate-osion alta on mahdollista tutkia prosenttilukuja siitä, kuinka useat käyttäjät määrättyltä aikaväliltä tulevat takaisin sovelluksen pariin tietyn

ajanjakson jälkeen. Retention Rate-tilastoja voidaan tutkia niin päivä-, viikko, kuin kuukausikohtaisestikin.

Rolling Retention puolestaan pitää kirjata aktiivisista käyttäjistä halutun ajanmääreen jälkeen. Kuten Retention Ratea, myös Rolling Retentionia voidaan tutkia niin päivä-, viikko, kuin kuukausikohtaisesti. Muita Flurry Analyticsin retentio-työkaluja ovat Sessions Per User sekä Recent Users, joiden avulla saadaan selville käyttäjän sessioiden määrä tietyin ajanjakson aikana, sekä viimeaikaisten käyttäjien määrä eri ajanjaksoilta.

Rolling Retention



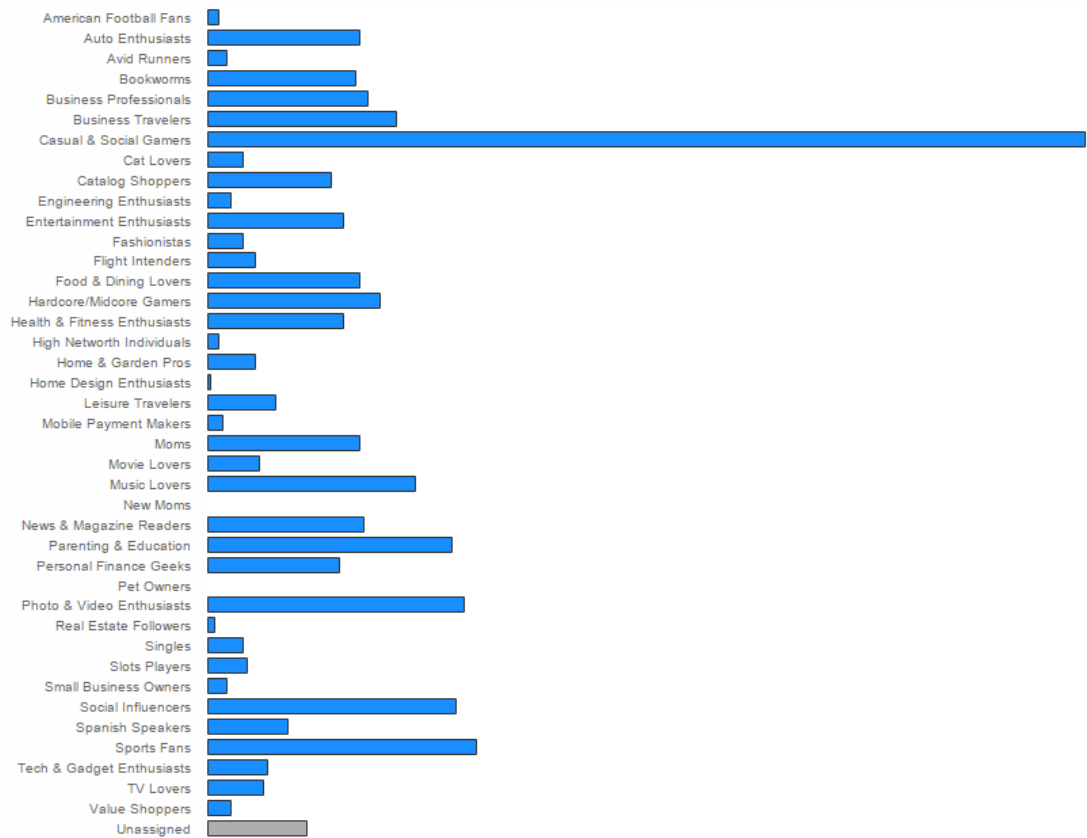
Kuva 8. Kuvankaappaus Retention-välilehden Rolling Retention - osiosta (Kyy Games Oy, Flurry, 2015, muokattu)

#### 2.4.4 Audience

Flurry Analytics pitää sovelluksen käyttäjäkunnasta kertovia tilastoja välilehden Audience alla. Käyttäjien intressejä voidaan tutkia omien muiden kehitettyjen sovellusten yhteisten käyttäjien kesken, sekä käyttäjien suosituimpien sovelluskategorioiden mukaan. Audience-välilehden alta löytyy myös Flurry Analyticsin Personas, joka on kehitetty näyttämään millaiset ihmiset käyttävät sovellusta. (Kuva 9) Välilehden alta löytyy myös tietoa käyttäjien maantieteellisestä sijainnista sekä käytetystä kielestä.

## PERSONAS

### Users by Persona - Percent of Audience



Kuva 9. Kuvankaappaus Flurry Personas - osiosta (Kyy Games Oy, Flurry, 2015, muokattu)

### 2.4.5 User Acquisition

Flurry User Acquisition Analytics (Flurry UAA) on suunnattu Flurryn palveluita käyttävien mainostajien työkaluksi. Tällä työkalulla voidaan luoda kampanjoita sekä tarkastella niiden kannattavuutta Flurry Analyticsin työkaluja käyttäen. (Flurry, 2015)

### 2.4.6 Events

Yksi Flurry Analyticsin tärkeimmistä ominaisuuksista, ellei jopa tärkein on Events-, eli tapahtumat-välilehti, jonne tulevien tapahtumien implementointiin keskityttiin opinnäytetyön aiemmassa luvussa. Events-välilehden alta löytyy yhteenveto tapahtumista, jossa listataan kaikki Flurrylle lähetetyt tapahtumat. Tältä sivulta voidaan myös valita tapah-

tuma tarkempaa tutkimista varten, jolloin voidaan tarkastella paremmin Flurryn luomia kaavioita, käyttäjien käyttäytymistä kyseisen tapahtuman jälkeen, tapahtuman parametreja sekä ajastetussa tapahtumassa sen kestoja.

Events-välilehden alta löytyy myös Funnels, eli suppilot. Suppiloita käytetään silloin, kun halutaan tutkia kuinka usein jokin tutkittavana oleva sarja tapahtumia tapahtuu. Suppiloita voidaan myös rajoittaa segmenteillä, jolloin voidaan tarkastella vain jotain tiettyä osaa käyttäjäkunnasta.

#### **2.4.7 Errors**

Mikäli sovelluksen Flurry Analytics-implemентаaatiossa on toteutettu virheiden raportointi, ilmaantuneita virheitä voidaan tarkastella tämän välilehden alta. Flurry Analyticsin avulla voidaan pitää kirjaa ja tutkia sovelluksen kaatumisia sekä tapahtuneita virheitä niiden nimen ja stack tracen mukaan.

#### **2.4.8 Technical**

Technical-välilehdeltä löytyy Flurry Analyticsin keräämät tiedot niistä laitteista, niiden käyttöjärjestelmistä sekä niiden operaattoreista, joissa sovellusta on käytetty. Technical-välilehdeltä löytyy myös osio, jossa on nähtävillä sovelluksessa ilmenneet virheet, jotka Flurry Analytics on onnistunut taltioimaan.

#### **2.4.9 Manage**

Flurry Analytics-projektin hallinnointi tapahtuu Manage-välilehdellä. App Info-osion alta löytyy projektin nimi, alakategoria, ominaisuudet, ADomain mainostuksen tarkkailuun, tämän hetkinen Flurry Analytics SDK versio sovelluksessa, sovelluksen ID kaupassa, sekä sovelluksen Flurry Analytics Application Key. Versions-osiolta löytyy kaikki sovelluksen eri versiot, joihin Flurry Analytics on implementoitu. Segments-osiossa listataan kaikki luodut segmentit ja voidaan luoda uusia. Events-osiossa näkyy lista kaikista eri tapahtumista, joita sovelluksesta on kutsuttu ja niitä voidaan joko suo-

dattaa tai poistaa kokonaan, jolloin niiden kutsuminen ei tee enää mitään. Alerts-osiossa näytetään ja voidaan luoda hälytyksiä kun jotain määrättyä tapahtuu. Permissions-osion alta löytyvät kaikki projektiin liitetyt käyttäjät. Advanced-osion alla on mahdollisuus poistaa sovellus Flurry Analyticsista.

### 3 HAVAINNOT EDELLISESTÄ FLURRY ANALYTICS-INTEGRAATIOSTA

#### 3.1 Yleistä

Cabals: Legends on korttipeli, joka perustuu pakan rakentamiseen. Itse pelissä käydyt taistelut tapahtuvat automaattisesti, joten pakan rakentaminen on pelissä ainoa tapa vaikuttaa taisteluun niiden valitsemisen lisäksi. Peliin voi hankkia lisää kortteja ostamalla korttipakkauksia, joista tulee korttipakkauksen tason mukaisia kortteja. Samanarvoisia kortteja voidaan myös yhdistää kovemman tason korteiksi, jolloin niistä tulee parempia. Itse pelissä maailmankartalle (Kuva 10) ilmestyy tapahtumia, eli pääosin taisteluita, joissa pelaaja taistelee korttipakkansa avulla vihollisia vastaan. (Kuva 11)



Kuva 10. Cabals: Legends karttanäkymä (Kyy Games Oy, 2014)



Kuva 11. Cabals: Legends taistelunäkymä (Kyy Games Oy, 2015)

Valtaosassa Kyy Games Oy:n pelejä, kuten myös Cabals: Legendsissä, käytetään free-to-play-monetisaatiomallia, jossa tärkeimpinä asioina pidetään pelaajan välitöntä kouluttamista peliin, pelaajien määrän kasvattamista ja etenkin ylläpitämistä sekä ostokäyttäytymistä. Tästä johtuen pelin analytiikkatoteutuksen tuli keskittyä tutkimaan erityisesti aivan pelin alkua, sitä koska pelaaja esittää oireita pelistä lähtemisestä, sitä koska pelaaja lähtee lopullisesti pelin parista, sekä ostokäyttäytymistä ja kuinka näihin tilanteisiin päädyttiin.

### 3.2 Mitä haettiin

Pääasiallinen tiedon kerääminen toteutettiin tapahtumien avulla, joita sisällytettiin pelin tärkeimpiin tilanteisiin. Kriittisimmistä tapahtumista luotiin myös versiot joita kutsuttiin vain silloin kun pelaaja laukaisi tapahtuman aivan ensimmäistä kertaa. Tämä siksi, että saataisiin parempaa ymmärrystä siitä, mihin asti pelaajat pääsivät kussakin pelisessiossa. Pelaajan ensimmäisen session tarkasteluun kiinnitettiin muutenkin eniten huomiota, jotta saataisiin paras kuva siitä, pääsikö pelaaja helposti peliin sisälle vai koituiko jokin tilanne liian ongelmalliseksi.

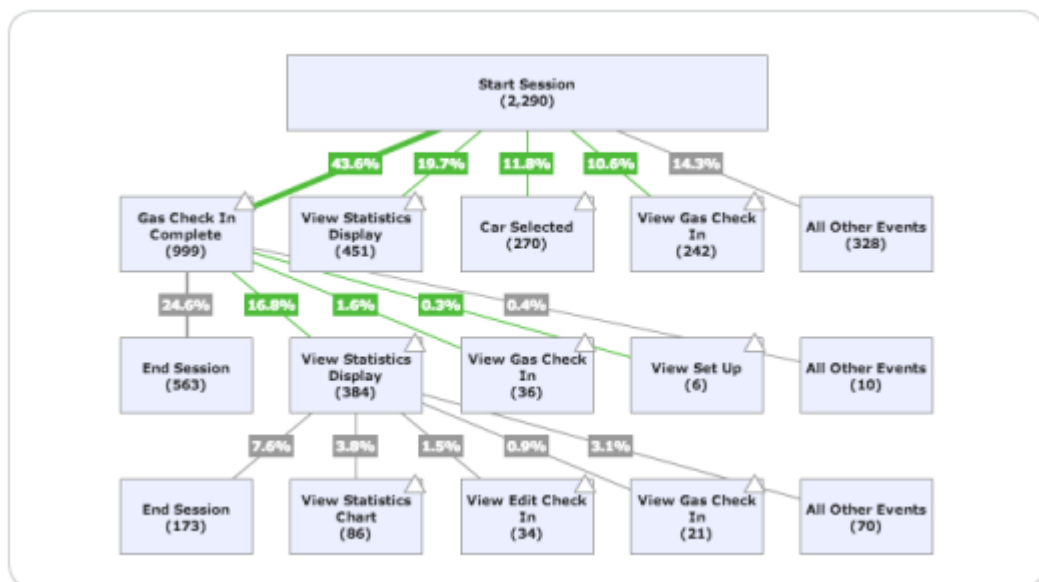
Käyttäjien liikkuminen pelin eri tilojen välillä ja tapahtumat eri tilojen sisällä olivat tärkeitä kerättäviä tietoja. Pelin sisällä liikkuminen ja liikkumattomuus kertovat niin navi-

goinnin selkeydestä kuin mahdollisista siihen liittyvistä ongelmista. Lisäksi haluttiin tietää pelaajan sen hetkinen sekä aiemmat sijainnit, jotta voitaisiin ongelmien ilmetessä saada parempi käsitys niiden alkuperästä.

Haluttiin myös saada tietoon mahdollisimman paljon siitä, miten pelaaja pelaa, joten parametreja käytettiin surutta hyödyksi. Yhteen parametriin listattiin niin paljon tietoa kuin vain Flurry Analyticsin rajoitusten mukaan voi.

### 3.3 Miten siinä onnistuttiin

Tapahtumien sijoittelu oli toimiva ratkaisu ja siitä saatiin haluttua tietoa irti. Ensimmäistä kertaa tapahtuvien tapahtumien hyödyntäminen oli hyvä ajatus ja sen avulla saatiin hyvää tietoa käyttäjän ensiaskeleista pelissä. Flurry Analytics tarjoaa tähän mainion työkalun, Events-välilehdeltä löytyvän User Paths-osion, jonka työkalun avulla pystytään silmäilemään näppärästi käyttäjien kulkua pelissä. (Kuva 12) Sen avulla tapahtumista muodostetaan polku ja voidaan tutkia kuinka moni käyttäjä pysyi määrätyllä polulla ja kuinka moni erkani siltä.



Kuva 12. Kuvankaappaus Flurry Analyticsin User Paths-osioista. (Flurry, 2015)

Parametrien kanssa onnistuttiin niin hyvin, kuin huonostikin. Jotkut parametrit toimivat varsin hyvin ja kertoivat selkeästi kaiken tarvittavan. Sen sijaan osaan parametreja sisällytettiin aivan liikaa tietoa ja niiden tulkitsemisesta tuli käytännössä mahdotonta ilman

pääsyä Flurry Analyticsin tietokantaan. Parametrien jaottelu pelaaja- ja tapahtuma-parametreihin oli silti hyvä ja toimiva ajatus, jota tullaan varmasti hyödyntämään myös jatkossa.

### 3.4 Havaitut ongelmat

Flurry Analytics-portaali on ollut usein hyvin hidas toiminnaltaan, mikä johtunee suuresta globaalista käyttäjäkunnasta. Ongelma on hieman lieventynyt Flurryn siirryttyä Yahoo Mobile Developer Suiten alle.

Tapahtumien-, tapahtumiin sisällytettävien parametrien- sekä parametrin sisällön merkien maksimimäärä tulee melko helposti vastaan, kun pyritään keräämään mahdollisimman laajasti tietoa pelaajan laukaisemista tapahtumista. Tällä hetkellä Flurry Analyticsiin voi sisällyttää 300 eri tapahtumaa, joille voidaan antaa 10 eri parametria, joille taas voidaan antaa korkeintaan 255 merkin pituisia syötteitä. Flurry Analytics-portaalissa myös näytetään vain 500 yleisintä parametriarvoa. Mikäli siis halutaan numerista tietoa pelaajasta niin tietojen yhtenäistäminen jollain keinolla saattaa joissain tapauksissa olla tarpeen. Yksi vaihtoehto tähän on muuttaa arvot ennalta määrättyjen raamien sisälle, esimerkiksi ryhmiin, joiden määrää voidaan kontrolloida ja joiden lukeminen Flurry Analyticsin portaalista on helpompaa.

Suurin ongelma itse Flurryn puolelta, johon törmäsi etenkin ensimmäisten Flurry Analytics-implementaatioiden aikana oli tiedon hidas saattaminen analysoitavaan muotoon Flurryn palvelimilla. Lähetetty tapahtuma saattoi tulla analysoitavaan muotoon vasta seuraavana päivänä. Tämä oli todella harmittavaa, etenkin tilanteissa joissa olisi tarvittu nopeasti tieto toteutuksen toimivuudesta.

## 4 MITEN PARANTAA FLURRY ANALYTICS-INTEGRAATIOTA

### 4.1 Mitä halutaan tietää

Alkuasetelmat riippuvat osittain pelin monetisaatiomallista, mutta jokaisella pelillä on tiettyjä yhteisiä asioita, jotka ovat samoja monetisaatiomallista riippumatta. Pelit ovat pohjimmiltaan viihdettä ja viihteen tarkoituksena on tarjota kuluttajalleen mielihyvää. Vaikka käyttäjän tuntemuksia ei sinällään voida kosketusnäytön kautta muuntaa binäärimuotoon, voidaan oikein sijoitetuilla tapahtumilla ja parametreilla saada selville pelin kohdat, jotka aiheuttavat pelaajalle negatiivisia tunteita tai ylitsepääsemättömiä ongelmia. Kun ongelmakohdat on löydetty, voidaan tehdä suunnitelmat ja sitä myötä toteutukset, joilla ongelmakohdat voidaan ratkaista.

Tavoitteena on aina saada käyttäjä mahdollisimman hyvin peliin sisään ja herättää kiinnostusta, joten aivan pelin alku ja ensimmäiset sessiot tulee miinoittaa tapahtumilla vielä entistäkin paremmin. Tämä korostuu etenkin silloin, jos pelissä on tutoriaali, jonka avulla pelaajaa ohjataan peliin sisään. Tutoriaalın tarkoitus on herättää pelaajan mielenkiinto ja opastaa pelin perusteet, eikä häätää pelaajaa pois pelin parista. Mikäli valtaosa pelaajista tippuu pelistä pois tutoriaalın aikana, lienee parasta tehdä jotain, mutta ilman tutoriaalın miinoitusta tapahtumilla tähän tietoon ei välttämättä pääse käsiksi.

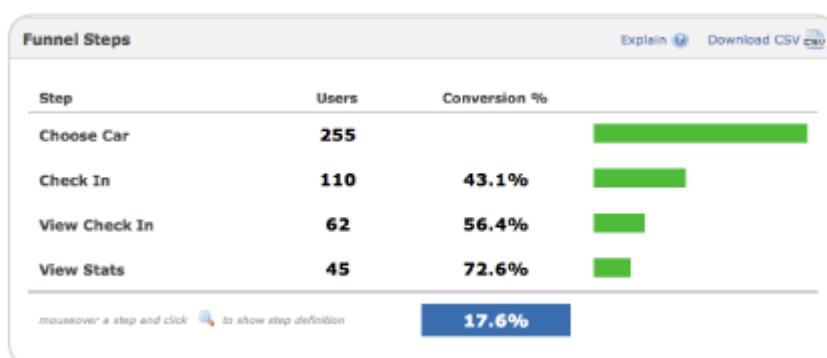
Pelistä pois jääviä pelaajia on tärkeä seurata muutoinkin, jotta voidaan löytää pelaajien ns. churn point. Churn pointilla tarkoitetaan sitä hetkeä, jolloin aiemmin aktiivinen pelaaja lopettaa pelin pelaamisen. Mikäli churn point voidaan edes jossain määrin ennustaa, pystytään kehittämään keinoja millä pelaaja saataisiin pidettyä edelleen mukana pelissä. Tämä ei ole tehtävä aivan helpoimmasta päästä ja Flurry Analytics jää hyvin helposti vajavaiseksi tiedon keräämisen suhteen. Hyvinkin tarkka ennustaminen ei ole silti aivan mahdotonta. Tästä hyvänä esimerkkinä mainittakoon Venäläinen MMO-julkaisija Innova, joka tutki pelaajien churn pointtia NCSofitin Aionin venäläisessä versiossa. He onnistuivat saavuttamaan 95 % tarkkuuden ennustaessaan pelaajien churn pointin kahdesta kolmeen viikkoa ennen sen tapahtumista. (Nozhnin, D. 2012)

Pelaajien ostokäyttäytyminen on myös eräs mielenkiintoinen ja tärkeä osa-alue, johon analytiikkaa käytetään. Free to play – peleissä pyritään ohjaamaan käyttäjää kaupan

puolelle joko pienillä nyansseilla tai lähes pakottamalla. Näiden eri menetelmien tehokkuutta on hyvä tutkia analytiikan avulla ja valita niistä se, joka toimii parhaiten.

## 4.2 Miten siihen päästään

Cabals: Legendsiin toteutetussa Flurry Analytics-implemmentaatiossa toteutettiin tapahtumat ja niiden ensilaukaisut varsin onnistuneesti, joten sitä voidaan hyödyntää myös jatkossa etenkin ensisession tutkimisessa. Tapahtumat tulee tietysti sisällyttää kulloinkin kyseessä olevan pelin kulun mukaisesti sopiviin paikkoihin, jotta saataisiin paras mahdollinen kuva pelisession etenemisestä. Käyttäjän ostotapahtumiin tulisi myös sisällyttää tapahtumia, jotta voidaan luoda funneleita, eli suppiloita tai tutkia muulla tavoin sitä, miten hyvin erilaiset ärsykkeet toimivat ostotapahtuman laukaisemisessa. (Kuva 13)



Kuva 13. Kuvankaappaus Flurryn funnel-esimerkistä (Flurry, 2015)

Parametrien käyttö on hyvin hyödyllistä ja kannattavaa, mutta parametrien arvojen kanssa ei kannata lähteä liiallisuuksiin. Cabals: Legendsissä yritettiin saada käyttäjältä lähes koko pelitilanne tietoon, mutta Flurry Analyticsia käytettäessä se todettiin käytännössä mahdottomaksi. Parhaiten tutkittavia ja luettavia parametreja saadaan silloin, kun hyvin suurella skaalalla esiintyvät luvut jaotellaan jo valmiiksi kutsuttaessa eri ryhmiin. Mikäli lukuja voi tulla välillä 0-1000, niin luvut voidaan jaotella ryhmiin esimerkiksi 200:n välein. Tällöin 0-200 olisi pienimpien lukujen ryhmä kun taas 801-1000 olisi suurimpien lukujen ryhmä ja niin edelleen. Tästä jaosta saatavaa viiden ryhmän kokonaisuutta on helppo tutkia Flurry Analytics portaalissa.

Käytettäviä funneleita, eli suppiloita tulisi suunnitella jo valmiiksi ennen implementaation aloittamista, jotta tapahtumat voitaisiin asettaa niiden suhteen suotuisiin paikkoihin.

Funnelit ovat oivallinen apuväline kun halutaan tutkia tapahtumien kulkua esimerkiksi: “Kuinka moni käyttäjä siirtyy kauppaan hävityn taistelun jälkeen?” -tyyppisillä kysymyksillä.

Erialaisten käyttäjäkuntien erittelyyn tulisi käyttää segmenttejä, joita voidaan sittemmin hyödyntää esimerkiksi funneleiden apuna, kun tutkitaan miten erinäiset käyttäjäkunnat eroavat totumuksiltaan pelin sisällä. Luotuja segmenttejä voidaan käyttää lähes kaikissa Flurry Analytics portaalin toiminnoissa, joten ne ovat hyvin yleishyödyllisiä kun halutaan tutkia jotain tiettyä asiaa eri käyttäjäkuntien suhteen.

### **4.3 Miten vältetään ongelmat**

Peliä kehittäessä tai jatkokehittäessä juuri mikään ei ole varmempaa kuin jatkuva muutos ja iterointi. Jotta analytiikkatoteutus pysyy iteroidessa mukana, on hyvä miettiä mahdollisia eteen tulevia ongelmakohtia jo etukäteen. Analytiikan ei tule olla vain julkaisuvalmiin pelin päälle liimattu palikka, josta pusketaan tietoa valitulle palvelimelle, vaan integraalinen osa pelikehitystä. Sen implementoimiseen tulee kiinnittää paljon huomiota heti pelin suunnittelusta lähtien, jotta pelin ongelmakohdat löydetään mahdollisimman nopeasti ja että ne voidaan korjata.

Flurry Analytics-toteutusta suunnitellessa kannattaa pitää tapahtumien määrä minimisissä, että voidaan varmistaa mahdollisten tulevien tapahtumien lisääminen jälkikäteen. Kaikkia kolmeasataa tapahtumaa ei tarvitse käyttää heti alkuun, vaan parametreilla voidaan hoitaa paljon asioita. Parametreihin ei kuitenkaan kannata yrittää sisällyttää koko peliä, vaan tärkeimpiä ja harkittuja tietoja kustakin tilanteesta. Tämän lisäksi tulee myös muistaa, että parametritkin tulee suunnitella huolella, ettei niiden kanssa tule jatkossa ongelmia ja että niistä saadaan mahdollisimman suuri hyöty.

Tärkeintä on kuitenkin huolehtia siitä, että analytiikkatoteutus palvelee pelille asetettuja tavoitteita ja vastaa siihen liittyviin ja sille esitettyihin kysymyksiin. Analytiikkatoteutuksen ei tule vain tuottaa hirvittävää määrää tietoa, vaan tuottaa sellaista tietoa, josta saadaan analysoimalla oikeasti hyödyllistä tietoa irti.

## 5 JOHTOPÄÄTÖKSET JA POHDINTA

Tiedon kerääminen ja analysointi ei kuulu enää pelkästään suurimpien pelialan yritysten repertuaariin, vaan kokoluokasta riippumatta kaikkien pelialan yritysten tulisi kiinnittää huomiota analytiikkaan. Tämä on entistä tärkeämpää, mikäli yritys julkaisee pelejään ensin ns. soft launchina, jonka aikana pelejä hiotaan entistä parempaan kuntoon ennen täyttä julkaisua. Tällöin tiedonkeruu, kerätyn tiedon analysointi ja siitä ilmenneisiin ongelmakohtiin reagointi on vielä kriittisempää, kun pyritään pelin parhaaseen mahdolliseen lanseeraukseen.

Flurry Analytics on hyvin kenttätestattu järjestelmä, johon luottavat mobiilimaailmassa monet tunnetut kehittäjät ja julkaisijat. Uusia analytiikkaratkaisuja ilmestyy kuitenkin jatkuvasti lisää ja eri ratkaisut vastaavat eri kehittäjien tarpeita paremmin. Täysin pelikehitykseen suunnattuja ratkaisuja on myös tullut saataville ja ne tuovatkin paljon uusia ominaisuuksia, joita Flurry Analyticsiin ei ole mahdollista sisällyttää sen universaalien mobiilituotteen takia. Yksi tapa saada molempien maailmojen parhaat puolet on sisällyttää erilaisia analytiikkaratkaisuja samaan sovellukseen ja jaotella niiden käyttö haluttujen ominaisuuksien mukaan.

Mikäli Flurry Analytics tai muut analytiikkaratkaisut eivät täytä kaikkia tarpeita, voi olla hyvä ajatus pohtia omaa analytiikkaratkaisua. Tällöin pystyy itse määrittelemään rajat itse toteutukselle, jos palveluntarjoajien ratkaisuissa huomataan puutteita tai liian rajoittavia ominaisuuksia. Omat analytiikkaratkaisut ovat tarpeen myös silloin, mikäli kyseessä oleva peli on moninpeliin perustuva, jota pelataan erikseen ylläpidetyillä servereillä. Flurry Analytics ja monet muut ovat sinällään täysin asiakaspuolen ratkaisuja, joten niiden kautta ei serverianalytiikkaa voi saada selville.

## LÄHTEET

Flurry. About Flurry. Luettu 1.4.2015.

<http://www.flurry.com/about-flurry>

Yahoo Developer Network, Flurry Analytics. Advanced Features for Android. Luettu 1.4.2015.

<https://developer.yahoo.com/flurry/docs/analytics/gettingstarted/technicalquickstart/android/>

Flurry, Flurry Analytics. FlurryAndroidApplication.java. Luettu 1.4.2015.

<https://gist.github.com/flurrydev/fbcd01ccdc3fe0cc264a>

Luban, P. 2011. The Design of Free-To-Play Games: Part 1. Luettu 6.4.2015.

[http://www.gamasutra.com/view/feature/6552/the\\_design\\_of\\_freetoplay\\_games\\_.php?print=1](http://www.gamasutra.com/view/feature/6552/the_design_of_freetoplay_games_.php?print=1)

Flurry. Flurry + Yahoo: The Next Chapter. Luettu 1.4.2015.

<http://www.flurry.com/blog/flurry-insights/flurry-yahoo-next-chapter>

Flurry. Flurry User Acquisition Analytics. Luettu 12.4.2015.

[https://www.flurry.com/sites/default/files/resources/User\\_Acquisition\\_Analytics\\_Overview.pdf](https://www.flurry.com/sites/default/files/resources/User_Acquisition_Analytics_Overview.pdf)

Nozhnin, D. 2012. Predicting Churn: When Do Veterans Quit?

[http://www.gamasutra.com/view/feature/176747/predicting\\_churn\\_when\\_do\\_veterans\\_.php?print=1](http://www.gamasutra.com/view/feature/176747/predicting_churn_when_do_veterans_.php?print=1)