

Legends of Descent: Analytics in an Ad-supported Windows Phone Game

Ian Obermiller
Awesomer Games
ian@obermillers.com

Sam Bayless
Awesomer Games
s.m.bayless@gmail.com

ABSTRACT

Legends of Descent is an ad-supported roleplaying game on the Windows Phone platform. With over 128,000 downloads and thousands of game sessions per day, it provides a unique platform to analyze user gameplay patterns and behavior. Surprising insights around user sessions, advertising performance, and network connectivity were uncovered through analysis of data collected via custom instrumentation code. Six months after publishing world-wide, we revisit the original analytics framework and identify areas of the game's design that we would modify given the new information.

Categories and Subject Descriptors

D.2.7 [Software Engineering]: Distribution, Maintenance, and Enhancement – *enhancement*.

D.2.8 [Software Engineering]: Metrics – *product metrics*.

D.2.10 [Software Engineering]: Design and Features – *abstract data types, polymorphism, control structures*.

General Terms

Measurement, design.

Keywords

Mobile gaming, role-playing game, mobile analytics, Windows Phone, ad-supported game

1. INTRODUCTION

The world of mobile gaming is at once both easy to start in and difficult to succeed in for an independent game developer. One way to differentiate a game from the studio-produced, graphics-intensive paid games is to offer the game for free, but collect revenue by showing ads within the game. Because an ad-supported application is monetized in an entirely different manner than a one-time, up-front paid application, ongoing monitoring of user behavior is essential to getting the most revenue out of the app's users. In this paper, we describe the game in question and the analytics system implemented to collect data. From that data, we look at design decisions that will increase user retention and thus total game revenue. We also describe the changes we plan to make to the next version of the analytics code in order to collect more information about the users and their usage patterns.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiGames '12, August 13, 2012, Helsinki, Finland.

Copyright 2012 ACM 978-1-4503-1487-9/12/08... \$15.00.

2. LEGENDS OF DESCENT

About a month before Windows Phone officially launched in the U.S., we began working on a game for the new platform. *Legends of Descent* (LoD) [1] is a single-player, real time role playing game inspired by the Diablo [2] series. See Figure 1 for two screenshots of the game. LoD is written in C# using the XNA framework, and everything is randomly generated (levels, enemies, items, quests), allowing for a great deal of replay. With generated content, users get a different experience each time they play, and can advance their characters continually as the enemies become progressively more difficult. LoD was the #1 role playing game on the platform for 2 months following launch, and was featured in the Windows Phone Marketplace under free games on March 10th, 2012. The game also has a rating of 4 stars by 380 users and has over 128,000 downloads as of 3/25/12.



Figure 1. In-game screenshots

Our decision to go with the ad model (using Microsoft PubCenter Advertising) vs. paid model was driven by two factors. First, one of us had a number of games in the Marketplace already. The paid games grossed an order of magnitude (\$100s/mo) less than the free games with advertising (\$1,000s/mo). Second, *Legends of Descent* was never meant to be a commercial venture; the goal was to see if we could produce a popular game and really enjoy the process. To that end, free games typically see a vastly larger number of downloads than paid games. Other game developers have observed that when publishing both free and paid versions of the same game, the free versions garner up to fifty times the number of downloads [3]. Later in the paper we will revisit the decision to create an ad-supported game and look forward to the revenue model for the next version.

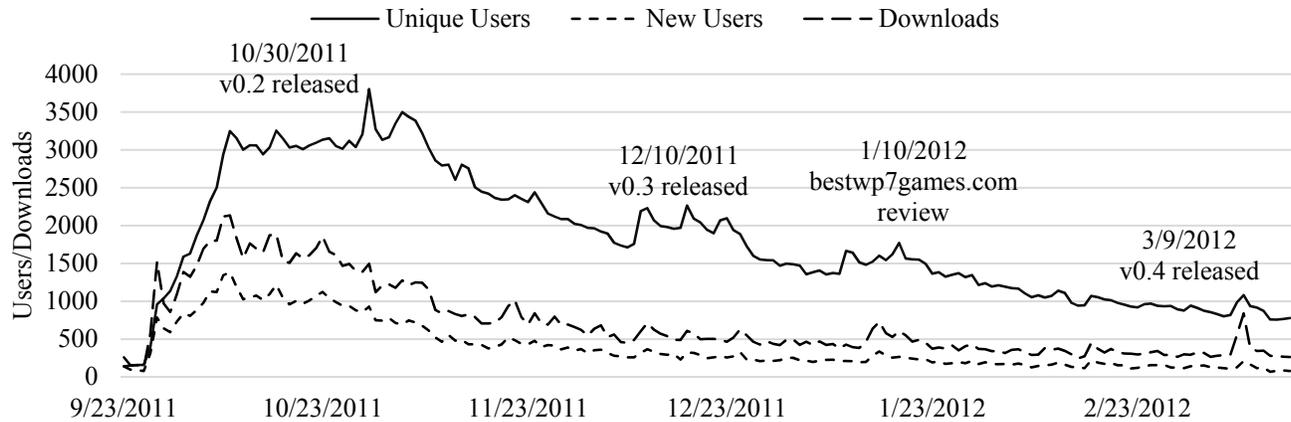


Figure 2. Timeline of users and downloads

2.1 Timeline

After launching the first version on September 22nd, 2011, we began releasing major content and gameplay updates:

- 9/22/2011 v0.1 - initial release
- 10/30/2011 v0.2 - quests, champion enemies, landscape mode
- 1/10/2012 v0.3 - player classes, blacksmith, and item stash
- 3/9/2012 v0.4 - boss enemy, town, and story line

A graph of users and downloads over the lifetime of the application, along with notations of important dates can be viewed in Figure 2. Notice the spikes in every line when a new version is released, or a signification review is posted. In any game, updates and marketing play a key role in sustaining a user base.

3. ANALYTICS

3.1 Implementation

Since the first release, LoD has been instrumented with simple analytics. Every time a user plays the game, their device id and anonymous user id, along with the times that the session started and ended are recorded and serialized to a compact binary format (~100 bytes) before being saved into persistent storage. The next time the game is started the analytics data is sent to our servers and pushed into a SQL database for further analysis.

3.2 Session

All data is collected around the concept of a *session*, which we define as an independent unit of time for which the game was played. To say a user has multiple sessions means that the user played the game multiple times. Session data is saved whenever an app is “deactivated” [4], the Windows Phone term for leaving the app. The game can be deactivated if the user hits the windows button, clicks on an advertisement, or gets a text message or phone call and chooses to reply or answer. Any reason for a deactivation will cause a new session to be created when the user returns to the game. Logically, two sessions separated by a short amount of time should be treated as a single session with a break in between. We addressed this issue during data processing by merging logs that were less than 30 seconds apart, recording the break duration and count. 42% of post-processed records had such

a break in them. In those records, the median break duration was 17 seconds, and the median number of breaks was 6, indicating that users are interrupted quite frequently while playing.

3.3 Alternatives

There is a plenitude of 3rd party analytics solutions for Windows Phone, including PreEmptive [5] (which was provided free for Windows Phone developers), Flurry [6], and Google Analytics [7], to name a few. We chose to use hand-coded, custom analytics in LoD mainly so that we would have unfettered access to the raw data, especially the device ids. This allows us to trace data back to each unique user. A user who has registered on our forums [8], for instance, can view the cumulative time they have played LoD and find out where they stand on the world-wide leaderboards by entering their device id.

3.4 Other Data Sources

Data about the game also comes from two other sources. The first is the Windows Phone AppHub, which provides us with a daily count of how many new users downloaded the app (excluding updates and reinstalls). The second is the Microsoft PubCenter Advertising analytics, which provides the total number of impressions, estimated revenue, and eCPM (estimated cost per impression) for each day.

3.5 Limitations

There are a few limitations with the way we collected data that make the analysis process more interesting:

1. Since data is only sent on launch, our analytics do not provide any information about users who only play once.
2. Since we only upload at the beginning of the session, if the user never has a network connection at that time we don’t get any data
3. As an oversight, time is only collected in UTC, and no time zone information was preserved.
4. The collected data points are limited; we only send the device id, anonymous user id, start time, and end time

As an example of our data deficiency, we know from the marketplace that our app has been downloaded 127,973 times (as

of Mar. 17, 2012), but we only have data for 72,040 unique devices. Although some of the downloaders may never have run the app that still leaves us with data for only 57% of downloaders.

4. DATA ANALYSIS

4.1 Users and Downloads

Figure 2 shows the unique users, new users, and downloads throughout the lifetime of the application, with important dates such as new version releases and reviews noted. Although we cannot compare the download count from the AppHub to the number of users we see in the analytics directly, we can look at the relative magnitude of changes from day to day. New users (those who were first “seen” by the analytics on that day) tracks downloads closely, as we would expect. The number of unique users per day, however, is more volatile. The spike on 10/30/2011, when v0.2 was released, indicates that the increase in unique users was due to existing users who saw the updated app, and being reminded about it, began to play again. We can infer this because a) downloads is relatively flat and downloads do not include updates, and b) the corresponding spike in new users is not nearly as drastic, compounded by the fact that data is only sent on the second run of the game.

This is an important point: updates are catering to previous users just as much, if not more, than to new users. Frequently updating a game with new content is not only a great way to keep existing players interested, but it is also a good method for helping old players remember your game so they can become interested again. Updates come with a cost, however, so the incremental value added with an update must be weighed against the time and resources taken to modify the game.

Next, we look at the graph of sessions per day (Figure 3), which shows a similar curve to Figure 2. The dark section of the graph represents the number of unique sessions per day, and is equivalent to the number of unique users in Figure 2. These numbers for sessions per day are consistent with other indie games for Windows Phone. Another game, for instance, was released in May 2011, and still receives 2,000-3,000 sessions per day [9].

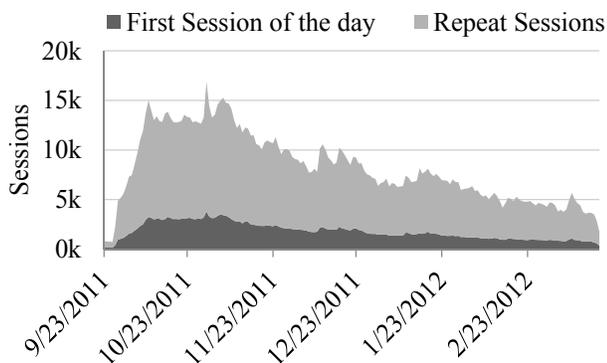


Figure 3. Sessions (game plays) per day

4.2 User Location

Though we did not collect user location information explicitly, we do get an indication of where users are from by the download information provided by the marketplace, which is broken down by country (Figure 4). Half of all downloads are from North America, a third are from Europe, and the remainder are from Asia and Oceania. This data suggests that our app might benefit from localization of the user interface and story line, to help draw in international players. Currently, all text is in English.



Figure 4. Downloads by country

4.3 Session Duration

By the very nature of the mobile platform, users are typically playing the game to fill gaps in time throughout the day. This is reflected strongly in our analysis (see Figure 5), with an average session length of 15 minutes. More indicative, 43% of sessions lasted less than 5 minutes, and 86% were under 30 minutes.

There are three primary methods we use to accommodate such short session durations. The first is that a user is able to pick up the game and immediately start playing with as little up-front direction as possible. Originally, LoD had a couple of long, text-filled screens that the player was expected to read the first time they played in order to understand the controls and the context of the world they were thrust into. After looking at the typical session length, though, we modified our approach to provide users with useful game information as it is needed. To start, for instance, a user is only told how to tap the screen to move.

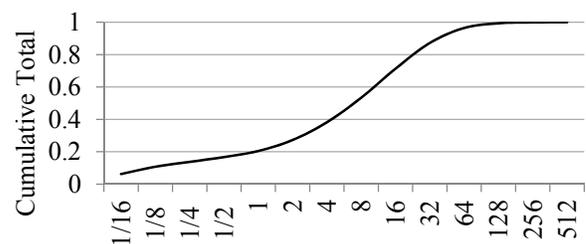


Figure 5. CDF of Session Duration (minutes)

As an incentive, the first quest is to walk around the starting town environment to find someone who will provide more directions. Upon finding the townspeople (who happens to be the quest giver), the user is shown how to attack (hold the button in the bottom left corner and tap the screen) and then told to exercise this new skill on a small invading horde of goblins. This sort of interactive tutorial is much less disruptive and allows the user to get into the interesting part of the game much quicker. Additionally, since implementing the tutorial, there have only

been two negative reviews complaining about the controls. The original negative reviews were likely due to people neglecting to read the previous long information screen.

The second method used to adapt to short sessions is to break up the game into small, manageable chunks. Gameplay in LoD consists largely of fighting one's way through a myriad of dungeon levels, one of which can be completed in an amount of time roughly equitable to the length of a typical session. This allows a player to feel that they accomplished something in the game even when they only had a short time to play. Finishing an entire level, however, is not always possible, as every player moves through the dungeon at a different pace.

The third method is to save the player's exact state whenever the app is deactivated. Any time that happens, the player's exact state is persisted to disk, including the current position in the level, all items on the ground, and all monsters, including their current status (alive, dead, wounded, etc.). When the application is "activated", any such state is immediately read from disk (or from memory if the OS decided to keep the app running), and the player is seamlessly returned to the point where they left off.

We next look at the average amount of time an individual user spends playing the game. Unique users are differentiated by their device id. Over the full time range, an average user plays 36 minutes per day. Intriguingly, as the game ages, the average amount of time a user spends playing LoD per day increases linearly (see Figure 6). Sessions per day (Figure 7) exhibits a similar linear increase. This is probably due to dedicated players who play for an extended period of time each day making up more and more of the user base as the influx of new users subsides.

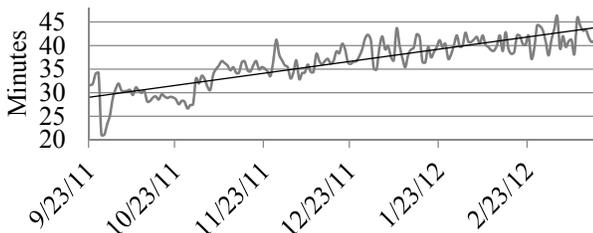


Figure 6. Average minutes played per user, per day

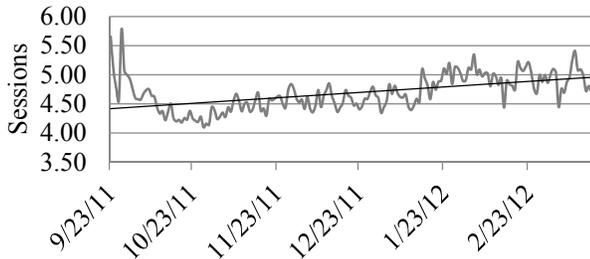


Figure 7. Average sessions per user, per day

4.4 User Retention

Figure 8 shows a CDF of total sessions per user over the lifetime of their gameplay (defined by the first and last times they appear in our analytic data). From the graph we see that the average user

plays the game 10 times before we stop getting data on them. Recall that we do not get data for users until the second session, so if we see them at all, they have played at least twice. The data varies per user however, with 31% only playing twice, 64% playing up to 5 times, and 80% playing up to 10 times. Similarly, the average user plays for a total of 2.5 hours, with only 50% playing longer than 15 minutes total and 29% playing longer than an hour. Finally, the average user plays for 17 days total, with 53% of users playing for 2 days or less (these numbers are defined by the difference between the first and last sessions we have on record).

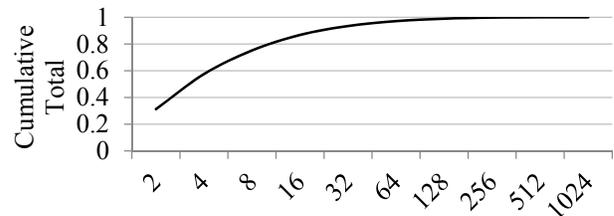


Figure 8. CDF of Total Sessions per User

These numbers have significant impact on the design and content of the game. Since half of the users quit after playing only a few times and for only 15 minutes, the first impression is often the last, so great content in the beginning of the game is essential to converting a casual user testing out the game into a regular player. For an ad-supported game, though, the long tail of players is extremely important and it is not enough to just capture the type of audience who plays a game for a short period of time and then quits. This is in contrast to a traditional paid game, where only the initial purchase earns revenue. In-app purchases bring a similar dynamic as the ad-supported model, as both rely on sustained playtime from both new and existing users.

Since LoD uses random generation techniques to create dungeons, choose the theme and enemies, and even for quests and item drops, there was originally no consistent gameplay for different users. To ensure the quality and variety of the core gameplay experience, we took out some of the randomness at the beginning of the game by hard coding the themes, enemies, and quests for the first 20 levels. This allowed us to control the player's exposure to more difficult enemies as well as providing a consistent story to follow through the game via the quest line.

4.5 When Do Users Play

Since the collected data did not include time zone information, we normalize all times to UTC-6 (US Central Time) since 50% of sessions come from the US and Canada. When plotting sessions per hour of the week, as in Figure 9, notice that the minimum value on the y-axis is 6,000; although the majority of users are in North America, the other half are playing at every hour of the day in every time zone. Indeed, such a result is indicative of the worldwide popularity of *Legends of Descent*. Per the graph, session count per hour peaks around 12pm, 3pm, and 7pm. Based on these times, we could infer that users typically play during a lunch break, during a mid-afternoon lull, and possibly during a commute or just downtime. The implication for game design is

that we must cater not only to the user looking for short and quick entertainment, but also to the user who wants to focus on playing the game. This is also supported by the session duration CDF in Figure 5.

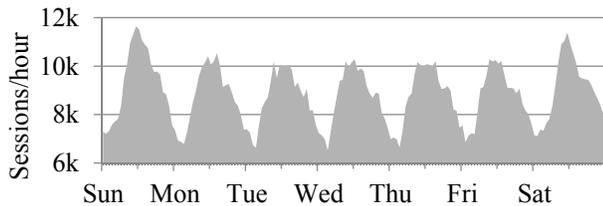


Figure 9. Sessions per Hour of Week

Sessions and downloads per day of week (Figure 10) peak on the weekends, and sessions seem to start the weekend early, with Thursday and Friday being elevated from the rest of the weekdays. This is also consistent with other mobile games [9].

Another aspect of retention is user engagement. Through our forums [8] and Facebook page [10], users can report bugs, request features, and socialize with other players. The Facebook page, as of 3/25/12, has 198 “likes” and 194 of comments posted on updates, photos, and videos. The forums (running a heavily customized version of mesoBoard), meanwhile, have replaced our Facebook page as the main source of news and community. As of 3/25/12, 117 registered users have made 2,071 posts in 320 threads on the forums. The most popular boards are Feature Requests and Bug Reports, indicating that the players participating are genuinely interested in the success of the game. The community has also provided us with a valuable group of beta testers, who provide feedback on game mechanics and balance while writing detailed bug reports in return for getting advanced access to the next version of LoD.

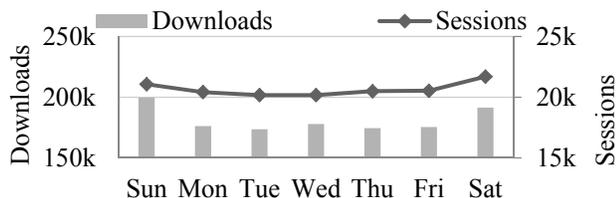


Figure 10. Downloads and Sessions per Day of Week

4.6 Advertising

Since *Legends of Descent* is a free, ad-supported game, 100% of revenue comes from advertising. We currently show two types of ads at the top center of the screen. 90% of the time, advertisements are served from Microsoft Advertising [11], and 10% are from AdDuplex [12]. AdDuplex, our alternate ad publisher, is chiefly an ad-swap provider. We show ads for other games in exchange for impressions for our own game. LoD gets approximately 100-150 click-throughs per day from AdDuplex.

Ads from Microsoft are all pay-per-impression, and they are the only ads we get direct revenue from. Around the same time that v0.2 of LoD was released, Microsoft Advertising began showing ads in new markets. In the game’s advertising code, when a request for an ad fails, we requested a new ad immediately to try

to fill the space. Unfortunately, the new markets almost always failed to provide an ad, and thus the number of ad requests ballooned. For their accounting, Microsoft reports impressions *requested*, not *served* (see Figure 11). So, even though impressions increased over 5x, the number of unique users had not significantly increased, and as a result the eCPM (effective cost per thousand impressions [13]) for each ad dropped considerably and revenue remained relatively unchanged. With this information, we will be modifying the advertising code to fall back to another provider instead of constantly requesting new ads.

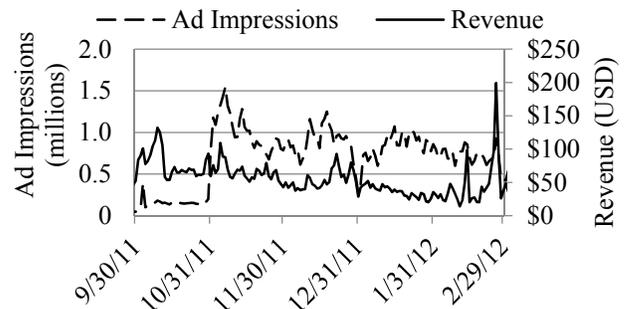


Figure 11. Ad Impressions and Revenue (Global)

4.7 Network Connectivity

The vast majority (98.55%) of uploads contained 10 or fewer sessions, and over 50% of uploads contained only one session. From this data we can infer that users of LoD have an active network connection ~47% of the time when they start the game.

Recall that the algorithm for collecting session data called for writing the session out to disk and only removing it after it was successfully uploaded to the server. This method will result in multiple sessions being batched together before being sent. Although the analytics server-side code did not record how many sessions were sent in a single upload, we were able to infer the value because subsequent rows inserted into the database would have continuously incrementing ids. Since inserts are done in parallel, this is not a foolproof method, but as the data shows it was able to detect a fair amount of batched uploads.

Even more telling, recall that 43% of users downloaded the game but never sent us any data (see 3.5) because they played the game only once, or not at all. If we assume some of those users actually played, but never had network access when playing, then it is possible that users in general have internet even less than 47% of the time *when playing the game*.

Such infrequent network access was truly surprising. It has a large implication on advertising, because if the user does not have network access, no ads will be shown. In light of this data, we’ll be modifying the advertising code to fall back to static resources advertising our other games if no ads can be downloaded. That way, the advertising space is not wasted.

5. LESSONS LEARNED

As detailed in section 3.5, there are several limitations with the current analytic code in *Legends of Descent*. We’ll be addressing these problems with the following changes.

First, we will send a subset of the data as soon as it is available (while the app is still running) in order to capture data about a user's first run. The remaining data will be sent on the next run.

Next, the data collected per session will be expanded. We will collect the app version in order to track usage of old versions and uptake of new ones. We will generate a unique id for each session for more accurate deduplication (previously, we were only able to remove sessions whose start and end times were equivalent for a single user), and additionally collect the startup mode, battery status, network mode, device information (firmware, manufacturer, etc.), total memory, locale, and time zone offset. We will not collect user location, because that requires adding the location capability, which notifies the user and activates the GPS, draining power. Instead, we will settle for a more coarse grained analysis based on the locale, time zone, and externally facing IP address of the device (typically the IP of the carrier's GGSN).

We will also report any unhandled exceptions in the code. Although the Windows Phone AppHub currently provides this information, we are only shown a list of unique exceptions and a daily count of all exceptions. This doesn't tell us which version of the app the user was running or when the exception actually occurred. With more contextual information, e.g. network status, we'd be better able to detect the root cause of a particular exception, e.g. no network connection.

Additionally, we'll be collecting an entirely new set of data specific to our game. Every session will send up the current gamer profile, consisting of a list of all of the players on the phone, along with their statistics, items, and progress through the game. This data will allow us to track the effectiveness of our content and gauge how and where to add new content in the future. It also has the added benefit of providing an easy means for online competition, as the profiles can be publicly displayed and associated with the player's forum [8] identity.

We also plan to release a paid version of LoD. The ad-supported model has definitely served us well, and we will always have a free version of the game. The ads are somewhat intrusive, however, and if you are not careful they can be touched accidentally. The paid version will only differ from the free in that the ads have been removed. In the future, we may offer more paid features in the form of in-app purchases, such as additional items, levels, or other content.

Finally, based on our experience, we offer the following recommendations for mobile game developers:

- If your game can be played offline, it is more difficult to monetize, but will be played more often.
- Optimize content and gameplay for short sessions, including quick startup, an integrated tutorial, manageable chunks of gameplay, and curated randomness.
- Advertising – to compensate for flaky ad providers, especially for internationally targeted games, use multiple ad services and fallback to static ads when none are available.
- Analytics – collect as much information as you can and send it as soon as it is available.

6. RELATED WORK

Several other projects have collected data from the user's phone to drive optimizations or provide insight. MobiTrack [14] did so with a separate app running in the background to collect phone interaction, app usage, and contextual parameters. MyExperience [15] collected data in the background and used a local database periodically synced with a central server, with the goal of reducing power without degradation of user experience. Shye, et al. [16] also collect data in the background and send it periodically when a network connection is present. As our approach is targeted for a single application, and we want the analytics to be bundled with the game, having a separate collection app was infeasible. This is the most important difference between our approaches.

7. REFERENCES

- [1] Microsoft, "Legends of Descent," <http://www.windowsphone.com/en-US/apps/9e0d80d9-6f7b-4c37-808e-68e270dde4f0>.
- [2] Various, "Diablo (series)," [http://en.wikipedia.org/wiki/Diablo_\(series\)](http://en.wikipedia.org/wiki/Diablo_(series)).
- [3] E. Perez, "A Million Downloads!," occasionalgamer.com/2012/01/20/a-million-downloads/.
- [4] Microsoft, "Execution Model Overview for Windows Phone," <http://msdn.microsoft.com/en-us/library/ff817008.aspx>.
- [5] PreEmptive, "PreEmptive Runtime Intelligence Service," <http://www.preemptive.com/windowsphone7.html>.
- [6] Flurry, "Flurry Analytics," <http://www.flurry.com/product/analytics/>.
- [7] Google, "Google Analytics," <http://www.google.com/analytics/>.
- [8] Legends of Descent, "Legends of Descent Forums," <http://legendsofdescent.com>.
- [9] E. Perez, "Occasional Gamer," <http://www.occasionalgamer.com/gamestats/>.
- [10] Legends of Descent, "Legends of Descent on Facebook.," <https://www.facebook.com/pages/Legends-Of-Descent/110521525719792>.
- [11] Microsoft, "Microsoft Advertising PubCenter," <https://pubcenter.microsoft.com/>.
- [12] AdDuplex, "AdDuplex - Ad Exchange Network," <http://www.adduplex.com/>.
- [13] T. Chow, "eCPM -- what exactly is that?," 27 Feb 2006. <http://goo.gl/PIhuU>.
- [14] H. Verkasalo, "Analysis of Smartphone User Behavior," in *International Conference on Mobile Business*, 2012.
- [15] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison and J. A. Landay, "MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones," in *Mobisys*, Puerto Rico, USA, 2007.
- [16] A. Shye, B. Scholbrok and G. Memik, "Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures," in *MICRO*, 2009.