

MONITOROWANIE AKTYWNOŚCI UŻYTKOWNIKÓW W KONTEKŚCIE MONETYZACJI APLIKACJI MOBILNYCH

MAGDALENA KIERUZEL¹, MACIEJ SZOPIŃSKI²

¹ Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Wydział Informatyki

e-mail: mkieruzel@wi.zut.edu.pl

² Scrum Master

Home.pl

e-mail: maciej@szopinski.net

SŁOWA KLUCZOWE

monetyzacja, agregacja danych, wskaźniki monetyzacji, aplikacje mobilne

STRESZCZENIE

Niniejszy artykuł odnosi się do problemu zbierania informacji o zachowaniu użytkowników aplikacji mobilnych, a w szczególności tych zachowaniach, które przekładają się na konkretny wynik finansowy, czyli tzw. monetyzację. W artykule zaprezentowano podstawowe wskaźniki wspierające analizę zachowań klientów. Przedstawiono również rozwiązanie techniczne – narzędzie wspierające proces pozyskiwania danych o zachowaniu użytkowników. Proponowane rozwiązanie zbudowane zostało na podstawie silnika Unity 3D, który jest dziś podstawą na rynku rozwiązań mobilnych. Powodzenie Unity 3D wynika głównie z tego, że jest to narzędzie programistyczne, służące do wytwarzania aplikacji z poziomu edytora, a przy odpowiedniej konfiguracji pozwala na dodawanie kolejnych elementów aplikacji bez ingerencji w kod źródłowy. Dodatkowo w zaprezentowanym rozwiązaniu wykorzystano gotowy zestaw narzędzi i bibliotek programistycznych Django wraz z rozszerzeniem Django Rest Framework.

Wprowadzenie

Ilość dostarczanych na rynek aplikacji mobilnych można dziś mierzyć w milionach wydań, co dla ich twórców stanowi poważne wyzwanie. Finansowy sukces związany z pozytywnym przyjęciem produktu dotyczy tylko niewielkiej ilości pozycji. Wsparciem dla komercyjnego sukcesu staje się ciągły monitoring zachowań odbiorców, kontakt z użytkownikami, śledzenie trendów oraz najlepsze dopasowanie propozycji do oczekiwań klientów. Producenci niejako

zmuszeni przez rynek gromadzą dane o klientach, o tym, w jaki sposób korzystają oni z proponowanych pozycji, jaka jest częstotliwość odwiedzin oraz kiedy najchętniej dokonywane są zakupy w aplikacji. Gromadzenie tego typu danych jest już podstawą w planowaniu i rozwoju istniejących rozwiązań mobilnych, a bez tego budowanie kolejnych wydań staje się bardzo ryzykowne – podatne na duży wpływ czynnika losowego. Wszelkiego rodzaju nagrody, planowane przy korzystaniu z danej aplikacji, powinny być dostosowane do faktycznych realiów związanych z ich funkcjonowaniem. Jeśli na przykład wprowadzamy nagrodę za odwiedzinę w aplikacji wynikającą z trzydniowych logowań, musimy mieć podstawę, by sądzić, że takie przypadki w ogóle miały miejsce, w przeciwnym wypadku nagroda nie spełni oczekiwań. Śledzenie i agregacja danych statystycznych w sposób świadomy przez twórców daje oczywiste korzyści, jednak nie dla wszystkich ten temat jest podstawą funkcjonowania na rynku aplikacji mobilnych.

1. Rozwiązania wspierające monitorowanie aktywności użytkowników aplikacji mobilnych

Wśród istniejących rozwiązań wspierających gromadzenie i analizę danych, wymienić można przede wszystkim firmę Google i jej podstawowy produkt z tego zakresu, tj. Google Analytics. Dodatkowo po stronie konkurencji na uwagę zasługuje KISSmetrics. Wymienione rozwiązania nie są jedynymi dostępnymi. Dodatkowo wskazać można:

- GameAnalytics – rozwiązanie darmowe, wspierające takie technologie jak: iOS, Android, Unity3D oraz Xamarin,
- deltaDNA – rozwiązanie płatne 150 USD miesięcznie za 15 000 użytkowników, wspiera takie technologie jak: iOS, Android, Unity3D oraz GameMaker,
- Google Mobile App Analytics – rozwiązanie darmowe wspierające Google Analytics SDK,
- Flurry Analytics – rozwiązanie darmowe wspierające takie technologie jak: Objective-C, Android oraz Swift.

Jeśli chodzi o środowisko programistyczne, to tylko GameAnalytics oraz deltaDNA korzystają z silnika Unity3D. Jest to zintegrowane środowisko stanowiące obecnie podstawę dla tworzenia gier mobilnych. Rozwiązania analityczne dla aplikacji mobilnych wymagają w większości ręcznej implementacji w kodzie źródłowym. W konsekwencji prowadzi to do zwiększenia prac programistycznych i kosztów wytworzenia oferowanego produktu.

O tym, jak ważna jest analiza aktywności użytkowników aplikacji, świadczy chociażby to, jak bardzo ten temat jest wspierany przez czołowe firmy, takie jak Google czy Yahoo. Proponowane przez nie produkty z tego zakresu są dziś podstawą na rynku aplikacji internetowych.

2. Podstawowe wskaźniki aktywności użytkowników aplikacji mobilnych

Komercyjny sukces aplikacji mobilnej powinien przekładać się na konkretne wyniki finansowe związane z pozyskaniem środków od użytkowników. Temat ten związany jest z monetyzacją, czyli przełożeniem aktywności odbiorcy na wynik w postaci złożonej płatności bądź wyświetlenia zamieszczonej reklamy. Dlatego też rozpoznanie preferencji użytkownika oraz

tę, w jaki sposób chętniej wydaje pieniądze w ramach aplikacji i czy w ogóle jest skłonny do tego typu działań, staje się podstawą prognozowania przychodów twórcy, a więc jego szans na finansowy sukces. Na rynku aplikacji mobilnych występują obecnie dwie podstawowe grupy. Pierwsza grupa to aplikacje będące kanałem do dystrybucji rzeczywistych produktów (ang. *transactional apps*), takie jak Amazon, Booking.com czy Uber, zwane aplikacjami transakcyjnymi. Drugą stanowią aplikacje, które same w sobie są produktem (ang. *product apps*), takie jak Facebook, WhatsApp czy Twitter. W niniejszym opracowaniu zaprezentowane zostaną wskaźniki efektywności dedykowane aplikacjom z drugiej grupy.

Podstawowe informacje, które można uzyskać, odnoszą się po pierwsze do aktywności użytkownika, a po drugie do wspomnianej monetyzacji, przekładającej się na przychód po stronie twórcy. Wydobycie informacji dotyczącej aktywności użytkowników bazuje głównie na dwóch zdarzeniach, tj. uruchomieniu aplikacji przez użytkownika (tutaj ważna jest data i godzina zdarzenia) oraz zakończeniu korzystania z aplikacji (wraz z datą i godziną zdarzenia).

Wśród najpopularniejszych wskaźników wykorzystywanych przy analizie aktywności wymienić należy **retencję** (ang. *retention*). Badanie tego wskaźnika ma szczególne znaczenie w darmowych aplikacjach mobilnych, przede wszystkim darmowych grach, gdyż tutaj liczy się zbudowanie długoterminowej relacji z użytkownikiem. Zadowolony, zaciekawiony użytkownik chętniej powraca do gry i jest skłonny w przyszłości zapłacić za nowe doświadczenia wynikające z użytkowania aplikacji. Dlatego też gry powinny wykazywać wskaźnik wysokiej retencji, co pozwala na zbudowanie relacji z odbiorcą. Retencję oblicza się na podstawie liczby użytkowników powracających do aplikacji. Dzień pobrania jest tzw. dniem zerowym (tutaj nie liczymy retencji), dopiero kolejne odwiedziny w aplikacji w dniu następnym świadczą o „zatrzymaniu” użytkownika i wtedy liczona jest retencja. Wskaźnik ten budowany jest dla różnych czasokresów (liczonych od dnia instalacji), najczęściej dla dnia, tygodnia lub miesiąca.

Kolejnym ważnym wskaźnikiem jest liczebność aktywnych użytkowników. Wskaźnik ten możemy stosować dla:

- dni – liczba aktywnych użytkowników w danym dniu **DAU** (ang. *Daily Active Users*),
- tygodni – liczba aktywnych użytkowników w danym tygodniu **WAU** (ang. *Weekly Active Users*),
- miesięcy – liczba aktywnych użytkowników w danym miesiącu **MAU** (ang. *Monthly Active Users*),

Zastosowanie wymienionego wskaźnika pozwala na obserwację zmieniającej się bazy użytkowników danej aplikacji. Na podstawie liczby aktywnych użytkowników widzimy, jaki rozmiar ma nasza baza i jaka jest tendencja – wzrostowa czy malejąca. W praktyce wskaźnik ten łączony jest z analizą retencji, co wydaje się zasadne z punktu widzenia nie tylko obserwacji ilości użytkowników nowych i powracających, ale również ich późniejszego zaangażowania po stronie aplikacji mobilnej.

W analizie i zbieraniu danych istotne wydają się intensywność oraz powtarzalność w korzystaniu z aplikacji. Pomocne jest tutaj badanie stosunku liczby użytkowników aktywnych w skali

jednego dnia do liczby użytkowników aktywnych w skali jednego miesiąca. Odzwierciedleniem tej proporcji jest wskaźnik **DAU/MAU**.

W przypadku aplikacji mobilnych istotną rolę odgrywa **pierwsze wrażenie użytkownika** (ang. *First Time User Experience – FTUE*). Badanie tego zjawiska jest ważne, ponieważ od pierwszego wrażenia najczęściej (przy dużej konkurencji na rynku) zależą dalsze losy aplikacji. Badana jest tutaj historia zachowań odbiorców, cały proces od instalacji programu po wykonanie w nim choćby początkowych aktywności (np. zapoznanie z instrukcją czy lektura samouczka). Późniejsza analiza tychże danych może posłużyć do wyciągnięcia wniosków na temat produktu. Jeżeli zainteresowanie (liczba początkowych pobrań aplikacji) było duże, ale i liczba późniejszych porzuceń równie wysoka, to analiza przyczyn wydaje się być oczywista. Powody mogą być tutaj różne, począwszy od błędów w projekcie, po niewielką użyteczność proponowanego rozwiązania.

3. Podstawowe wskaźniki monetyzacji aplikacji mobilnych

Przy badaniu zjawiska monetyzacji interesuje nas zysk, jaki aplikacja jest w stanie wygenerować dzięki wydatkom użytkowników. Jak wynika z raportu SWRVE z roku 2016, tylko 0,23% graczy, a więc niewielka liczba, dostarcza aż 60% przychodu w skali miesiąca. Konieczna jest tutaj analiza pozostałej części odbiorców i określenie możliwych działań zmieniających oszczędne zachowania klientów. Twórcy aplikacji mobilnych najczęściej monetyzują swój produkt poprzez wprowadzanie opłat za korzystanie z aplikacji. Występują opłaty jednorazowe lub tzw. stała opłata subskrypcyjna. Dodatkowo, w celu zwiększenia dochodu, stosowane są zakupy w ramach aplikacji (ang. *In-App Purchase*) oraz reklamy. Użytkownicy najmniej chętnie wybierają rozwiązania z góry wymuszające odpłatność przed skorzystaniem z gry. Dlatego częściej twórcy aplikacji proponują tzw. mikropłatności – dotyczy to zakupu np. nowych funkcjonalności czy usprawnień. Mimo tak konstruowanych mechanizmów zakupów nie są one konieczne, aby wersja podstawowa aplikacji nadal działała.

Podstawowym wskaźnikiem mierzącym dochód generowany ze wszystkich transakcji w danej aplikacji jest **łączy dochód**. Kolejnym obserwowanym jest **transakcja**, czyli całkowita łączna liczba unikalnych transakcji w ramach sklepu zaimplementowanego w danej aplikacji.

Mając do dyspozycji dwa powyższe wskaźniki, możemy budować **średni dochód na transakcję**. Wskaźnik ten określa średnią wartość wydatków użytkowników na zakupy w aplikacji. Analiza tego wskaźnika daje podstawy do ewentualnych zmian w zakresie oferty i cen w sklepie.

Średni dochód na użytkownika wskazuje wartość przeciętnych wydatków, jakie pojedynczy klient przeznaczona na zakupy w aplikacji. Wskaźnik ten jest przydatny, jeśli prognozujemy zmianę w dochodach w zależności od liczby użytkowników.

Przydatnym wskaźnikiem jest **konwersja** obrazująca stosunek liczby tych klientów, którzy wykonali co najmniej jedną transakcję w formie gotówkowej, do całkowitej liczby klientów – użytkowników aplikacji w danym dniu (DAU).

Odsetek płacących użytkowników jest podobnym wskaźnikiem do poprzedniego – badamy tutaj całkowitą liczbę klientów którzy dokonali zakupu w stosunku do liczby wszystkich klientów danej aplikacji.

Analiza monetyzacji aplikacji mobilnych wspierana jest również przez obserwację **wartości czasu życia użytkownika** (ang. *Life Time Value, LTV*). Wskaźnik ten może być interpretowany jako wartość, którą klient generuje dla danej aplikacji. Szacowanie tej wartości odbywa się na podstawie średniego dziennego dochodu na użytkownika oraz liczby dni do ostatniego logowania w aplikacji.

Liczba odsłon reklam w aplikacji jest najbardziej istotna tam, gdzie monetyzacja opiera się głównie lub jedynie na reklamach zewnętrznych partnerów. Generowane raporty dotyczą tutaj głównie liczby kliknięć na reklamę, czyli zainteresowania ich treścią przez użytkowników.

Wybór i zastosowanie wskaźników jest dla twórców aplikacji rzeczą indywidualną, powyżej zostały zaprezentowane wskaźniki najczęściej wykorzystywane. Dodatkowym wsparciem dla analizy byłoby zautomatyzowanie wydobywania danych dotyczących użytkowników, jak i agregacji tych danych, bez konieczności ręcznej ingerencji w kodzie źródłowym aplikacji.

4. System automatycznego zbierania i analizy danych o użytkownikach aplikacji mobilnych

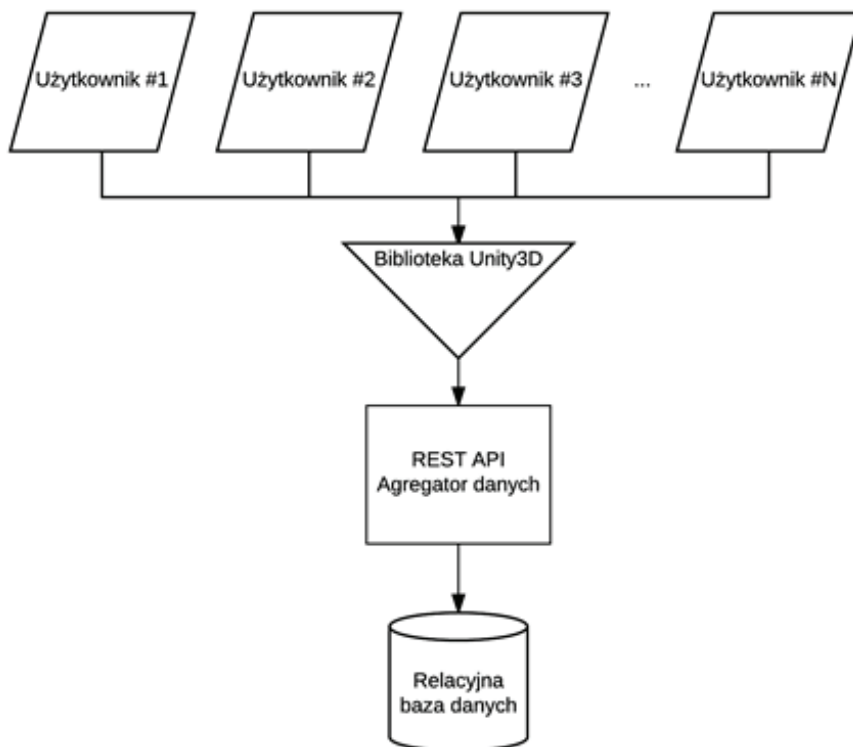
Dotychczas nie powstały mechanizmy analogiczne do rozwiązania Google Analytics wykorzystywanego w technologiach internetowych, gdzie proste wywołanie skryptu pozwala na śledzenie aktywności na witrynie internetowej. Narzędzia dostępne na rynku zwykle udostępniają użytkownikom **API** (ang. *Application Programming Interface*) umożliwiające przesłanie danych do chmury, jednak po stronie zespołu programistycznego pozostaje kwestia przygotowania odpowiednich danych i przesłanie ich na serwer. Opracowanie zautomatyzowanego mechanizmu, który nie wymaga dodatkowej ingerencji programisty w kod aplikacji, znacząco zwiększyłoby możliwości deweloperów. Praca w takim schemacie nie wydłuży czasu powstawania aplikacji, a umożliwi lepsze dopasowanie do użytkownika i w efekcie końcowym może zostać składową komercyjnego sukcesu aplikacji.

Rozwiązanie zaprezentowane w ramach niniejszego artykułu ma za zadanie w jak największym stopniu zautomatyzować proces zbierania danych o użytkownikach aplikacji mobilnych. Za cel postawiono zaimplementowanie biblioteki programistycznej, która w sposób kompleksowy i w jak największym stopniu automatyczny pozwoli na zbieranie danych statystycznych w aplikacjach mobilnych. Ułatwi to w znacznym stopniu twórcom aplikacji rozpoczęcie procesu zbierania i analizy danych, które w przyszłości mogą stanowić bezpośrednie przełożenie na kierunek dalszego rozwoju. Pamiętając, że jest to narzędzie w rękach deweloperów, rozwiązanie oparte zostało na jednym z najpopularniejszych na rynku silników umożliwiających tworzenie aplikacji i gier mobilnych, Unity3D. Narzędzie stworzone przez Unity Technologies może pochwalić się największym w tym momencie procentowym udziałem na rynku gotowych narzędzi wspierających pracę zespołów programistycznych (47% zgodnie z raportem Developer Economics: State of the Developer Nation). Co ważne, jest to udział większy niż tworzenie aplikacji

z wykorzystaniem natywnych mechanizmów udostępnianych przez producentów danego systemu. Silnik popularność zyskał dzięki możliwości kompilacji jednej bazy kodu i konfiguracji na 21 dostępnych platformach sprzętowo-systemowych. Rozszerzenie możliwości silnika o agregację danych o użytkownikach trafi w efekcie do szerokiego grona użytkowników docelowych.

Zaproponowane rozwiązanie opiera się na dostępnych w ramach silnika Unity3D komunikatach informujących aplikację i obiekty w widoku o rozpoczęciu i zakończeniu działania. Dwie informacje, które w efekcie końcowym przekładają się na zdecydowaną większość zidentyfikowanych powyżej wskaźników monetyzacji i aktywności użytkowników. Zaimplementowana biblioteka silnika Unity3D reaguje na te zdarzenia i w sposób asynchroniczny komunikuje się z serwerem agregującym dane. Dla działania aplikacji jest to przezroczysty proces i nie powinien negatywnie wpływać na wydajność. Komunikacja zachodzi jednokierunkowo, z aplikacji mobilnej do serwera. Bez znaczenia jest w tym kontekście odpowiedź serwera. Warunkiem koniecznym do przesłania informacji z aplikacji do serwera jest dostęp do internetu w urządzeniu korzystającym z badanej aplikacji. Jeśli z jakiegoś powodu nie uda się przesłać danych (awaria biblioteki, brak dostępu do internetu), nie może to powodować błędów w działaniu aplikacji. Dla zapewnienia odpowiednio wysokiej wydajności zastosowany został mechanizm współprogramów (ang. *coroutines*). Efekty działania współprogramów widoczne są w aplikacjach z dynamicznie zmieniającym się obrazem, gdzie zachowana musi być odpowiednia liczba klatek na sekundę w wyświetlanym obrazie (gry mobilne, aplikacje z dużą ilością animowanych elementów). W praktyce zastosowanie współprogramów w implementowanej bibliotece oznacza, że w sytuacji, gdy przesyłanie danych do serwera potrwa dłużej niż jedna klatka animacji, proces zostanie zatrzymany i wznowiony w kolejnej klatce. Dla dalszego uproszczenia korzystania z biblioteki całość udostępniona zostaje w formie gotowego do wykorzystania w edytorze Unity3D prefabrykatu, z prostym interfejsem użytkownika umożliwiającym dodatkową konfigurację biblioteki. Czynności, które w efekcie końcowym pozostają po stronie twórcy, to konfiguracja biblioteki przez podanie unikalnego klucza dla aplikacji mobilnej. Klucz ten pozwala w kolejnych etapach na identyfikację, jakiej aplikacji dotyczą wpisy zebrane w ramach serwera pełniącego funkcję agregatora danych statystycznych, co jest istotne, jeśli jednocześnie monitorujemy użytkowników wielu aplikacji mobilnych.

Dane zbierane w ramach aplikacji mobilnych nie powinny być przechowywane lokalnie na urządzeniach użytkowników. W ten sposób deweloper nie miałby do nich dostępu na żądanie. Rozwiązaniem w takiej sytuacji jest konfiguracja i udostępnienie serwera, który pełnić będzie funkcję agregatora dla danych przesyłanych z poszczególnych urządzeń i zainstalowanych na nich aplikacji. Schemat pracy biblioteki Unity3D wraz z agregatorem danych zaprezentowano poniżej.



Rysunek 1. Schemat pracy biblioteki Unity3D

Źródło: opracowanie własne.

W związku z powyższym serwer musi być systemem o wysokiej dostępności, z krótkim czasem odpowiedzi i zdolnym do przetworzenia dużej ilości żądań przechowywanych w relacyjnej bazie danych. Wymagania te spełnia framework do tworzenia aplikacji Django, oparty na Pythonie. Zarówno wbudowane narzędzia frameworka, jak i jego rozszerzenia, zapewniają prostą ścieżkę do implementacji API – zapisywania żądań w bazie danych oraz zarządzanie serwerem z poziomu panelu administracyjnego. Framework Django doczekał się w ostatnim czasie wersji oznaczonej numerem 1.9 i posiada bogate wsparcie dla społeczności open source. Mnogość bibliotek i rozszerzeń znacząco usprawnia bieżącą pracę programisty. Dzięki zastosowaniu rozszerzenia o nazwie Django Rest Framework większość mechanizmów niezbędnych do uruchomienia API REST została automatycznie uwzględniona w aplikacji. Na potrzeby implementacji niniejszego rozwiązania Django umożliwiło zautomatyzowanie części prac przez wykorzystanie gotowych elementów pozwalających na projektowanie API czy zarządzanie obiektami w bazie danych dzięki wbudowanemu mechanizmowi ORM (mapowanie obiektowo-relacyjne).

W ramach serwera agregującego dane niezbędne było zaprojektowanie bazy przechowującej zdarzenia przesyłane z aplikacji mobilnych. Z punktu widzenia Django oznaczało to utworzenie modelu danych, dzięki któremu framework w sposób automatyczny generuje strukturę bazy danych w wybranej technologii (dostępne są m.in. MySQL, PostgreSQL, SQLite).

Technologia, w jakiej utworzony zostanie serwer agregujący dane, pozostaje tematem otwartym. Należałoby przeanalizować, czy Django, jako szeroki zestaw narzędzi, nie posiada zbyt dużego narzutu na tak prostą funkcjonalność jak API REST. Alternatywą pozostają mikroframeworki specjalizujące się w API, takie jak Silex czy Phalcon.

Zaimplementowana biblioteka, w połączeniu z serwerem agregującym dane, stanowi komplet otwierający drogę do analizy zachowań użytkowników i wskaźników monetyzacji aplikacji mobilnych. Z jednej strony zapewniona została spójność danych i mechanizm weryfikujący ich integralność, z drugiej aplikacja mobilna korzystająca z biblioteki silnika Unity3D nie traci pierwotnej wydajności. Wszystko dostarczone w formie wygodnego i gotowego do ponownego wykorzystania komponentu, stanowiącego jeden ze stałych elementów aplikacji.

Podsumowanie

Obserwacja zachowań użytkowników aplikacji nie może być tematem odległym dla ich twórców, głównie ze względu na silną konkurencję na rynku rozwiązań mobilnych. Oczywiście staje się więc zautomatyzowanie procesu pozyskiwania tego typu danych, jak i ich agregacji. Dziś wiele firm, takich jak Google czy Yahoo, widzi ten problem i udostępnia technologie wspierające twórców w zakresie pozyskiwania i analizy danych. Zaproponowane w artykule rozwiązanie jest dodatkową pozycją, opartą na popularnym środowisku programistycznym – Unity3D oraz Django. Dzięki opracowanemu systemowi możliwe stało się między innymi monitorowanie takich akcji jak uruchomienie i zakończenie działania aplikacji. Zdarzenia tego typu są podstawą przy badaniu aktywności użytkowników rozwiązań mobilnych. Jako dalszy rozwój warto byłoby wprowadzić monitorowanie takich akcji użytkownika jak zminimalizowanie aplikacji czy wczytanie / zmiana poziomu, co dałoby możliwość obserwacji długości przejścia danego etapu w grze. Mimo wskazanych propozycji rozwoju proponowane narzędzie wspiera optymalizację i automatyzację pozyskiwania danych o zachowaniu użytkowników, a więc może być z powodzeniem stosowane przy badaniu zjawiska aktywności klientów aplikacji mobilnych.

Literatura

- Fields, T., Brandon, B. (2012). *Social game design: monetization methods and mechanics*. Boca Raton: CRC Press.
- Gibson, J. (2014). *Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#*. Crawfordsville: Apress.
- Goldstone, W. (2011). *Unity 3.x Game Development Essentials*. Gliwice: Helion.
- Greenfeld, D. (2015). *Two Scoops of Django, Best Practices for Django 1.8*. Los Angeles: Two Scoops Press.
- Mele, A. (2015). *Django By Example*. Birmingham: Packt Publishing.
- Seif El-Nasr, M., Drachen, A., Canossa, A. (2013). *Game Analytics: Maximizing the Value of Player Data*, 13–40. Londyn: Springer-Verlag.

Seufert, E.B. (2014). *Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue*. Waltham: Morgan Kaufmann.

Thorn, A. (2014). *Pro Unity Game Development with C#*. Berkeley: Apress.

MONITORING OF USERS ACTIVITY IN THE CONTEXT OF MONETIZATION OF MOBILE APPLICATIONS

KEYWORDS | monetization, data aggregation, monetization metrics, mobile applications

ABSTRACT

Collection and analysis of data about users activity can not be overlooked by their creators, mainly due to strong competition in the market of mobile solutions. Today, many companies such as Google or Yahoo see that problem and provide technology to support developers in the field of data collection and analysis. The solution which is proposed in this article is undoubtedly an additional value for developers, based on the popular development tools, such as Unity3D and Django. Using this tool it is possible to monitor user activity within the application, from simple events such as starting and stopping the application to more complex actions. Events of this type are the basis of user activity research in mobile solutions. As a further suggestion, it would be helpful to examine actions, such as minimizing applications, load or level change, which would give the opportunity to observe the length of transition from one level to another within the game. The proposed tool supports the optimization and automation of data related with user behavior, so it can be successfully used for the study of user activity in mobile applications.

Translated by Magdalena Kieruzel and Maciej Szopiński

