Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Informatyki

Zegar do gier – przypominacz / budzik dla wielu zdarzeń o charakterze cyklicznym

Dokumentacja deweloperska

Damian Kudas, Michał Mrowczyk

5 czerwca 2013 r.

Zawartość

1. Wstęp i o autorach projektu

W niniejszym dokumencie opisujemy w jakim stopniu zamierzona i oczekiwana funkcjonalność produktu została zrealizowana. Szkicujemy również jakie elementy mogłyby zostać dodatkowo zaimplementowane i podajemy przykładowe pomysły na ich implementację. Nie opisujemy tu szczegółów implementacyjnych obecnej wersji produktu (szczegóły te zostały opisane w dokumencie: „Specyfikacja techniczna"). Również nie opisujemy tu sposobu korzystania z aplikacji z punktu widzenia użytkownika (ten temat został poruszony w dokumencie „Podręcznik użytkownika"). Zanim podejmiecie próbę kontynuacji projektu proszę przejrzyjcie również dokumenty: „Wizja" oraz „Koncepcja" zawierające krótki wstęp w problematykę projektu.

Zanim zdecydujecie się kontynuować nasz projekt, musicie wiedzieć, że dla osób go inicjujących była to pierwsza (i prawdopodobnie ostatnia) aplikacja napisana na platformę Android. Nie jesteśmy fanami GUI i tego typu wynalazków, więc nie znajdziecie u nas wyrafinowanych widgetów, a co najwyżej proste kontrolki nie sprawiające oszałamiającego wrażenia. Jeśli jednak wciąż chcecie podjąć nasz projekt i zmierzyć się z labiryntem przeszkód i pułapek zastawionych w świecie Javy i Androida to prosimy o dalszą lekturę…

2. Co zostało, co zostało częściowo, a co nie zostało zrealizowane…

Tutaj opowiemy pokrótce co zostało, a co nie zostało zrealizowane w aplikacji.

Opiszemy przykładowe rozwiązania które nie zostały zaimplementowane z powodu braku czasu, naszego lenistwa i innych czynników. Do przykładowych rozwiązań podamy pomysły na implementację.

Funkcjonalność, która została zrealizowana to:

- Możliwość tworzenia alarmów, wczytywania i zapisywania konfiguracji alarmów do pliku xml
- Możliwość definiowania parametrów alarmu (włączając w to: nazwę alarmu, nazwę gry dla tego alarmu, dźwięk alarmu, okres alarmu, głośność alarmu…)
- Możliwość usuwania po nazwie alarmu
- Możliwość przeglądania listy alarmów w GUI
- Możliwość zmiany okresu oraz godziny i minuty alarmu (wybierane w GUI). Zrealizowane za pomocą rysowania po Canvas'ie
- Synchronizacja alarmów (dbanie o to, aby godziny i minuty odgrywania alarmu były odpowiednio uaktualnianie w czasie działania przez aplikację) – korzystam z javovej klasy Calendar w tym celu
- Wypisywanie logów przy działaniu aplikacji np. o wynikach parsowania xml'a, akcjach użytkownika, informacji szczegółowych o alarmach itp.) Logi klasycznie tworzone są za pomocą System.out.println (co w przypadku Androida zapisuje na konsolę LogCata), czyli mamy do nich dostęp wyłącznie na etapie testowania na emulatorze z poziomu Eclipse'a

Funkcjonalność która została zrealizowana częściowo:

- Wypisywanie informacji o alarmach (nie zostało zrealizowane w stopniu całkowitym, bo użytkownik aplikacji powinien mieć dostęp do wszystkich informacji o alarmach w GUI, a nie tylko nazwy alarmu, inne informacje to np. czas za jaki alarm ma być odegrany lub gra do jakiej jest przypisany) Pomysł na realizację jest taki, żeby do ShowAlarmsActivity przekazywać jako parametr zserializowaną listę alarmów (zamiast listy nazw alarmów jak to jest obecnie) i wyciągać te dane – wymaga serializacji i jakiegoś pomysłu jak te dane prezentować na liście w ShowAlarmsActivity. Funkcjonalność nie została zrealizowana z powodu braku czasu i niechęci realizujących do używanej technologii…)

Funkcjonalność która nie została zrealizowana:

- Tworzenie customowych dźwięków z poziomu aplikacji przez użytkownika np. przez przypisanie nazwy dźwięku do ścieżki w systemie plików na Androidzie. Można dodać np. łatwy wybór dźwięku (np. jakiejś piosenki mp3) przez użytkownika. Uwaga: ta funkcjonalność będzie wymagała prawdopodobnie dorobienia nowego xmla i parsowania go – tym razem z dźwiękami customowymi. Główną motywacją dla wprowadzenia tej funkcjonalności jest ułatwienie użytkownikowi wyboru dźwięku alarmu przy tworzeniu alarmu. Obecnie jest to wczytywanie ścieżki (URI) do pliku z palca… Funkcjonalność ta nie została zrealizowana z powodów opisanych już uprzednio
- Możliwość dodawania / usuwania alarmu w czasie grania dźwięku alarmu – można to dorobić kopiując listę alarmów za każdym razem, żeby pozbyć się natrętnych

ConcurrentModificationException. Na obecnym etapie trzeba zatrzymać aktualnie działający alarm, aby tego dokonać.

- Jakiś bardziej wybuchowy interfejs graficzny. Powód niezrealizowania – brak umiejętności w zakresie, brak czasu spowodowany innymi zajęciami na uczelni, innymi projektami, kolokwiami, egzaminami oraz zainteresowaniami poza-informatycznymi twórców, wliczając w to życie osobiste.

3. Na co trzeba szczególnie uważać przy projekcie, czyli rzecz o bugach i niedoróbkach

Rys 1 – I love the smell of debugging in the morning …

Jeśli nie straszne wam bugi i niedoróbki, to dobrze trafiliście, bowiem autorzy tego projektu nie wiedzą co to OOP (Object Oriented Programming), a refaktoryzację traktują jak źródło wszelkiego zła. Ponadto fakt iż był to nasz pierwszy projekt w Androidzie nie pomaga, gdyż pewne rzeczy które chcieliśmy zrealizować w sposób A zostały zrobione w sposób: @#!#@! Taka kolej rzeczy była głównie spowodowana tym, że w trakcie implementacji przekonywaliśmy się że pewne rzeczy są niemożliwe do zrealizowania w sposób jaki założyliśmy i musieliśmy robić to w sposób, który nie chcieliśmy, aby nawet ujrzał światło dzienne.

Przykład 1:

Dodawanie i usuwanie alarmu nie jest zrealizowane w osobnych Activities albowiem twórca projektu na etapie realizacji tej funkcjonalności nie wiedział nawet o istnieniu Activities i

potem nie chciał zmieniać tego fragmentu z racji powszechnie wiadomych… (reguła: „działa – nie dotykaj”)

Przykład 2:

Hardcodowane stringi utrudniają podpięcie większej ilości języków (np. polski) do obsługiwanych przez naszą aplikację – póki co jest to tylko i aż język Szekspira.

Przykład 3:

Żeby korzystać z widgetu: ViewFlipper wrzucone zostały 3 layouty do jednego layoutu o nazwie: activity_main.xml

4. Kontakt

Jeśli będziecie realizować jednak ten projekt dalej, to w razie pytań, wątpliwości, uwag itp. proszę pisać pod adres: mrowczyk@student.agh.edu.pl