# Projekt

Celem projektu było napisanie aplikacji realizującą badania aktywności fizycznej w zakresie:

1. personalizację (waga, wiek, wzrost, płeć)

2. sensor

3. zbieranie danych

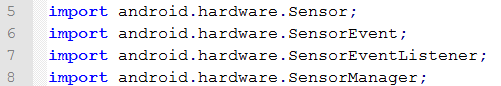
4. prezentację wyników (BMI lub siatki centylowe, dystans, czas aktywności)

5. eksport do plików zewnętrznych w celach statystycznych i archiwizacyjnych.

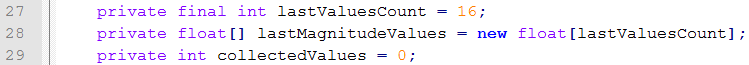
Aplikacja dostarcza podstawowe funkcjonalności pedometru bez konieczności użycia sprzętowego licznika kroków z zastosowaniem optymalnego algorytmu zapewniającego jej dokładność obliczeniową, stabilności oraz przenośność.

# Zasada działania aplikacji i przykład zastosowania

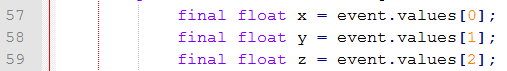
Test realizuje zadanie dzięki dostępnej funkcjonalności sensora telefonu oraz zaimportowaliśmy potrzebne biblioteki:



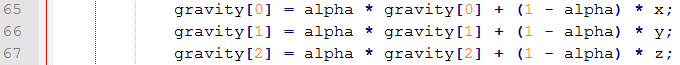
Krokomierz mierzy ilość kroków wykonanych przez użytkownika. Jest to proces bardzo szybki, bo w ciągu sekundy pobieranych jest około kilkanaście odczytów. Sensor pobiera szesnaście ostatnich odczytów a potem uśrednia wynik.



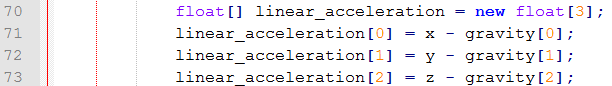
Po odczytaniu pomiarów



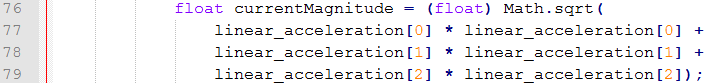
grawitacja jest odfiltrowywana w celu określenia rzeczywistego przyspieszenia urządzenia, wkład do siły grawitacji musi być wyeliminowane. Można to osiągnąć przez zastosowanie filtru *górnoprzepustowego*.



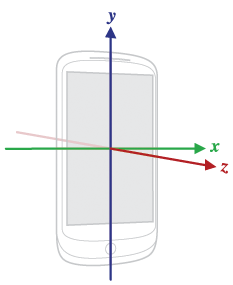
W wyniku dostajemy przyspieszenie liniowe.



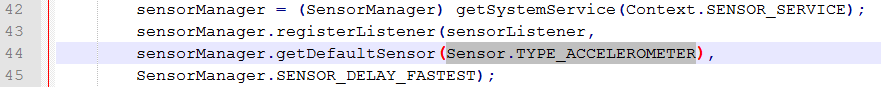
Wielkość wektora przyspieszenia obliczamy ze wzoru:



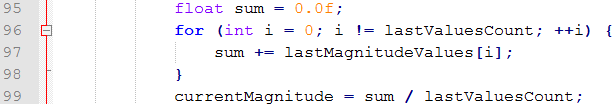
Układ współrzędnych jest zdefiniowany w stosunku do ekranu telefonu w jego domyślnej orientacji. Osie nie są zamienione, gdy orientacja ekranu urządzenia się zmienia.



Oś X jest pozioma wskazuje na prawo, oś Y jest ustawiony pionowo i skierowana w górę i punktów osi Z w kierunku na zewnątrz od przedniej powierzchni ekranu.

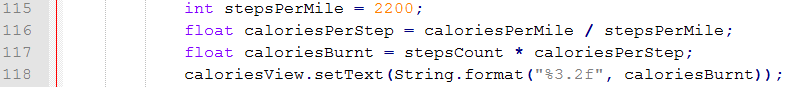


Uśredniamy lastValuesCount wyników, domyślnie 16 (jak wyżej).

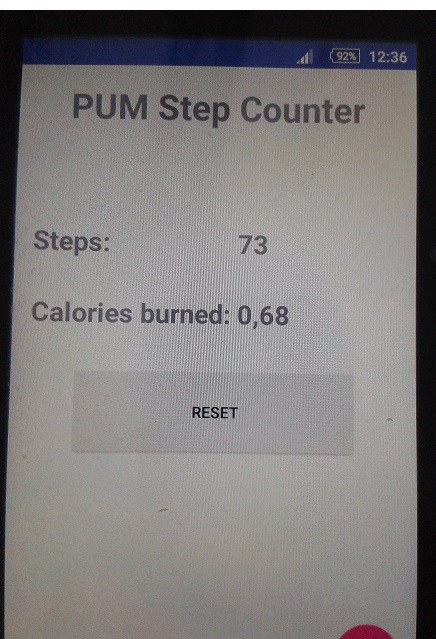


Z założenia dorosły mężczyzna wykonuje 2200 kroków na mile. Taki parametr w obecnej wersji (na cele testowe aplikacji) ustalono z poziomu kodu.

W dalszej kolejności jest możliwość włączenia dodatkowej funkcjonalności (w razie potrzeby poza okresem testowym).



Aplikacja posiada dwa ekrany, działa w tle i korzysta z czujnika ruchu. Ilość kroków prezentowana jest na bieżąco przez aplikację. Licznik można w każdej chwili zrestartować.



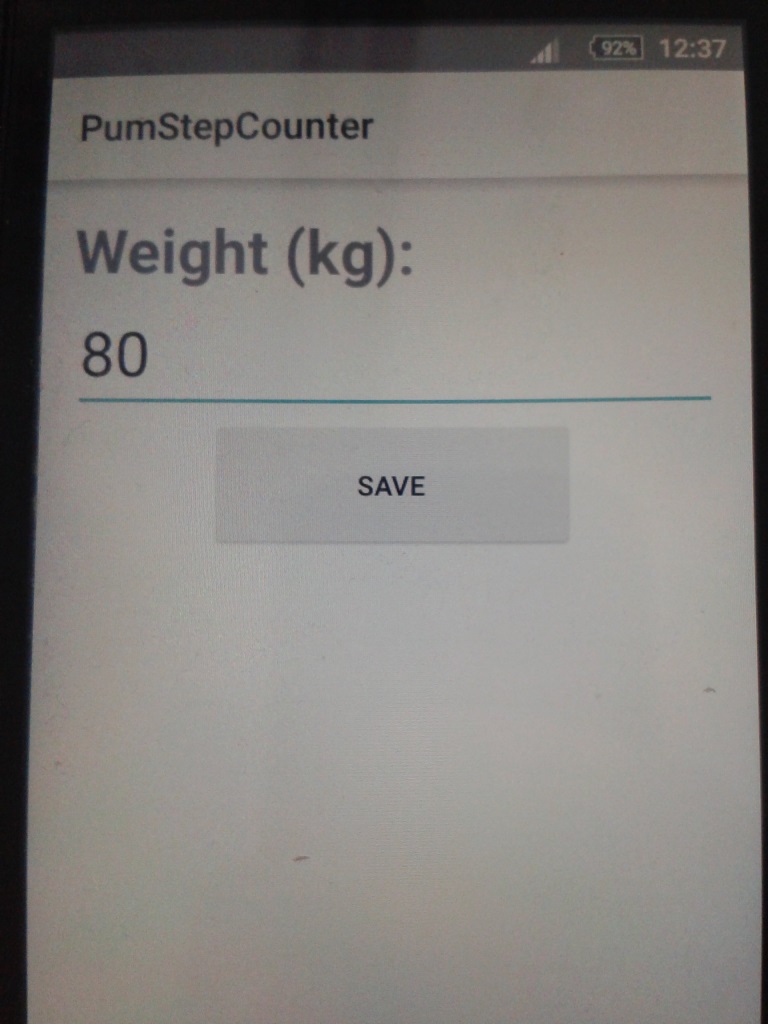
Rys. 1. Ekran główny aplikacji

Powyższy ekran przedstawia podstawowe

funkcjonalności aplikacji, takiej jak:

1. Zliczania kroków.
2. Licznika spalonych kalorii.
3. Możliwości restartu wyników.
4. Przejście do etapu ustawień.
5. PUM Step Counter personalizacjia (set-up)

Rys. 2. Ekran konfiguracyjny aplikacji



# Wnioski

Dzięki zastosowanemu algorytmowi udało się uzyskać zadawalającą precyzję ilości kroków będącą dostatecznie bliską stanu rzeczywistego. W ramach badania rozbieżności zastosowano metodę manualnych pomiarów a wyniki skonfrontowano z wartościami otrzymanymi ze sprzętowych liczników kroków oraz podobnych produktów konkurencyjnych. Dobór algorytmu i sensora ruchu był podyktowany optymalizacji precyzji pomiaru jak również optymalizacją współczynnika cena/jakość w kontekście zarówno użycia zasobów sprzętowych (w tym braku konieczności potrzeby włączonego GPS, dodatkowego osprzętu takiego jak armband). Zadawalającą precyzję pomiaru kroków uzyskano przez wybór do zarejestrowania sekwencji kroku akcelerometru, którego wyjście cyfrowe jest odpowiednio interpretowane przez oprogramowanie w drodze elektronicznej analizy liczby kroków podczas chodu lub biegu.

Poza realizacją zadań projektowych założonym celem aplikacji było pozytywne działanie na zdrowie jej potencjalnego użytkownika, jako urządzenia służące nie tylko samym pomiarom, ale też motywacji. Podczas testów aplikacji zauważono motywujące działanie pedometru spowodowane wydłużeniem codziennie przemywanego pieszo dystansu, jak również chętniejsze decydowanie się na przebycie drogi piechotą zamiast używania środka transportu zmechanizowanego.

# Literatura

1. <http://www.czasopismologistyka.pl/artykuly-naukowe/send/318-artykuly-na-plycie-cd-3/7022-artykul>
2. <http://web.cs.wpi.edu/~emmanuel/courses/cs403x/D15/slides/lecture06.pdf>
3. developer.android.com/reference/android/hardware/SensorEvent.html