APLIKACJE MOBILNE

GESTY PEŁNOEKRANOWE

dr Artur Bartoszewski

OBSŁUGA GESTÓW

- ✓ Do obsługi gestów w API Androida służy obiekt GestureDetector

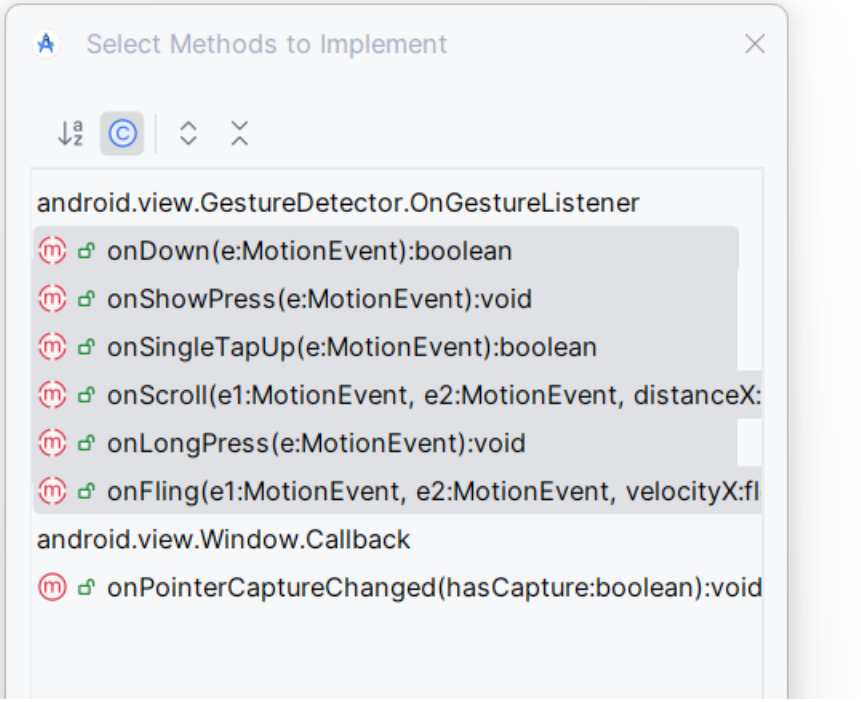
GestureDetector nazwa = new GestureDetector(kontekst, słuchacz);

- ✓ GestureDetector posiada interfejs OnGestureListener (słuchacz gestów), który implementuje metody odpowiedzialne za rozpoznawanie i obsługę różnego rodzaju gestów.
- ✓ Jedną z metod zaimplementowania słuchacza gestów jest rozszerzenie klasy MainActivity (głównej aktywności) o ten interfejs

```
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {
```

Implementacja metod listenera

```
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {
```



W tym przykładzie aktywność MainActivity implementuje interfejs GestureDetector.OnGestureListener.

Po zaimplementowaniu interfejsu należy uzupełnić aktywność o związane z nim metody

Implementacja metod listenera

```
@Override
public boolean onDown(@NonNull MotionEvent e) {
    return false;
}
@Override
public void onShowPress(@NonNull MotionEvent e) {
}
@Override
public boolean onSingleTapUp(@NonNull MotionEvent e) {
    return false;
}
@Override
public boolean onScroll(@Nullable MotionEvent e1, @NonNull MotionEvent e2, float distanceX, float distanceY) {
    return false;
}
@Override
public void onLongPress(@NonNull MotionEvent e) {
}
@Override
public boolean onFling(@Nullable MotionEvent e1, @NonNull MotionEvent e2, float velocityX, float velocityY) {
    return false;
}
```

Interfejs OnGestureListener implementuje wewnątrz klasy MainActivity metody obsługi gestów.

Implementacja metod listenera

1. onDown(MotionEvent e):
   - Wywoływana, gdy palec dotyka ekranu.
2. onShowPress(MotionEvent e):
   - Wywoływana, gdy palec dotyka ekranu i pozostaje w tym samym miejscu przez krótki czas (przed ewentualnym przesunięciem lub innym gestem).
3. onSingleTapUp(MotionEvent e):
   - Wywoływana, gdy palec dotyka ekranu i zostaje uniesiony (bez przesunięcia).
4. onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY):
   - Wywoływana, gdy palec przesuwa się po ekranie.
   - e1 to zdarzenie początkowe, e2 to zdarzenie końcowe.
   - distanceX i distanceY to odległości przesunięcia w poziomie i pionie.
5. onLongPress(MotionEvent e):
   - Wywoływana, gdy palec dotyka ekranu i pozostaje w tym samym miejscu przez dłuższy czas.
6. onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY):
   - Wywoływana, gdy palec przesuwa się po ekranie i zostaje szybko uniesiony ("rzut").
   - e1 to zdarzenie początkowe, e2 to zdarzenie końcowe.
   - velocityX i velocityY to prędkości przesunięcia w poziomie i pionie.

Dodawanie obiektu GestureDetector

```
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {

    GestureDetector gestureDetector;  1 usage

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        gestureDetector = new GestureDetector( context: this,  listener: this);

    }
```

Następnie w metodzie OnCreate tworzymy instancję słuchacza gestów – instancję klasy GestureDetector

- Drugim parametrem jego konstruktora jest „this” – czyli prościej mówiąc słuchaczem dla gestów staje się główna aktywność.

Przechwytywanie zdarzeń i przesyłanie do GestureDetector-a

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    detektorGestow.onTouchEvent(event);
    return super.onTouchEvent(event);
}
```

Aby przechwycić zdarzenie gestu nadpisujemy metodę: onTouchEvent(..)
Otrzymuje ona w parametrze obiekt event typu MotionEwent opisujący gest.

Zdarzenie to przesyłamy do zdefiniowanego wcześniej słuchacza gestów za pomocą metody onTouchEvent()

np.: detektorGestów.onTouchEvent(event)

Naciśnięcie ekranu - onDown

```
@Override
public boolean onDown(MotionEvent motionEvent) {
    float X  = motionEvent.getX();
    float Y  = motionEvent.getY();
    float xp =motionEvent.getXPrecision();
    float silaNacisku = motionEvent.getPressure();
    opis.setText("onDown: "+ String.valueOf(X)+":"+String.valueOf(Y)+
                 "\n"+String.valueOf(silaNacisku));
    return false;
}
```

Metoda reaguje na dotknięcie ekranu (natychmiastowo w momencie dotknięcia)

Obiekt „motionEvent" typu MotionEwent opisuje gest. Posiada on metody, za pomocą których odczytać możemy parametry gestu np.:

-.getX() .getY() - pozycja dotknięcia

- getPresure() – siła nacisku

„Tupnięcie” na ekran - onSingleTapUp

```
@Override
public boolean onSingleTapUp(MotionEvent motionEvent) {
    float X  = motionEvent.getX();
    float Y  = motionEvent.getY();
    opis.setText("onSingleTapUp: \n"+ String.valueOf(X)+":"+String.valueOf(Y));
    return false;
}
```

Metoda reaguje na krótkie stuknięcie w ekran.

Obiekt „motionEvent” typu MotionEwent opisujący gest posiada metody, za pomocą których odczytać możemy parametry gestu np.: .getX() .getY()

Krótkie naciśnięcie ekranu- onShowPress

```
@Override
public void onShowPress(MotionEvent e) {
    float px = e.getX();
    float py = e.getY();
    opis.setText("onShowPress " +
                 "\n x="+String.valueOf(px)+
                 "\n y="+String.valueOf(py));
}
```

Obiekt „e” typu MotionEwent opisujący gest posiada metody, za pomocą których odczytać możemy parametry gestu np.: .getX() .getY()

Długie naciśnięcie ekranu- onLongPress

```
@Override
public void onLongPress(MotionEvent motionEvent) {
    float X  = motionEvent.getX();
    float Y  = motionEvent.getY();
    opis.setText("onLongPress: \n"
                + String.valueOf(X)+":"+String.valueOf(Y));
}
```

Obiekt „motionEvent” typu MotionEwent opisujący gest posiada metody, za pomocą których odczytać możemy parametry gestu np.: .getX() .getY()

Przeciągnięcie po ekranie

Ges przeciągnięcia po ekranie generuje dwa zdarzenia:

- ✓ onScrool – zdarzenie ciągłe – na bieżąco reaguje na ruch
- ✓ onFiling – zdarzenie pojedyncze – wywoływane po przeciągnięciu po ekranie
- ✓ Do zdarzeń przekazywane są obiekty ***motionEvent*** (parametry początkowe zdarzenia) i ***motionEven1*** (parametry końcowe zdarzenia). Pozwalają one odczytać pozycję początku i końca przeciągnięcia oraz jego dystans (w rozbiciu na dystans po osi x oraz po osi y (prędkość gestu nie jest istotna)
- ✓ Parametry ***v*** i ***v1*** pozwalają odczytać prędkość gestu – w rozbiciu na prędkość poziomą i pionową

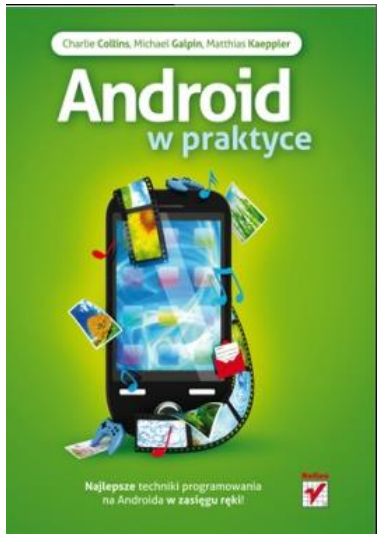
Przeciągnięcie po ekranie - onScroll

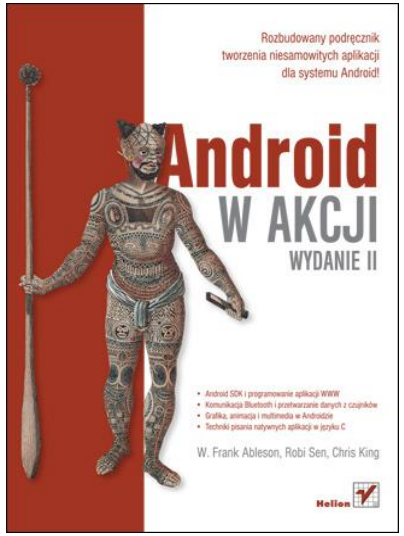
```
@Override
public boolean onScroll(MotionEvent motionEvent, MotionEvent motionEvent1, float v, float v1) {
    float X  = motionEvent.getX();
    float Y  = motionEvent.getY();
    float X1  = motionEvent1.getX();
    float Y1  = motionEvent1.getY();
    opis.setText("onSingleTapUp: \n"+ String.valueOf(X)+" : "+String.valueOf(Y)+"\n"
                +String.valueOf(X1)+" ; "+String.valueOf(Y1)+"\n"
                +String.valueOf(v)+" : "+String.valueOf(v1));
    return false;
}
```

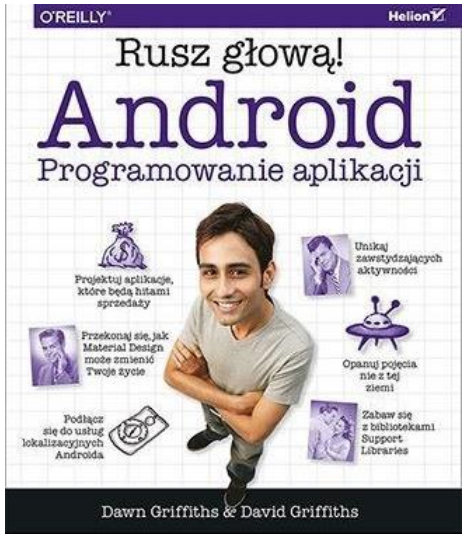
Przeciągnięcie po ekranie - onFiling

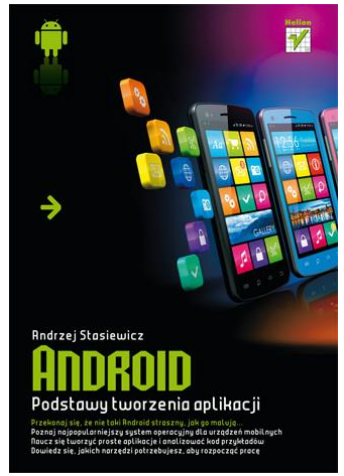
```
@Override
public boolean onFling(MotionEvent motionEvent, MotionEvent motionEvent1, float v, float v1) {
    float X  = motionEvent.getX();
    float Y  = motionEvent.getY();
    float X1  = motionEvent1.getX();
    float Y1  = motionEvent1.getY();
    opis.setText("onFling: \n"+ String.valueOf(X)+" : "+String.valueOf(Y)+"\n"
            +String.valueOf(X1)+" ; "+String.valueOf(Y1)+"\n"
            +String.valueOf(v)+" : "+String.valueOf(v1));
    return false;
}
```

Literatura


Charlie Collins, Michael Galpin, Matthias Kaeppler
Android
w praktyce
Najlepsze techniki programowania
na Androida w zasięgu ręki!


Rozbudowany podręcznik
tworzenia niesamowitych aplikacji
dla systemu Android!
Android
W AKCJI
WYDANIE II
W. Frank Ableson, Robi Sen, Chris King
Helion


O'REILLY
Helion
Rusz głową!
Android
Programowanie aplikacji
Projektuj aplikacje, które będą hitami sprzedaży
Unikaj zawstydzających aktywności
Przekonaj się, jak Material Design może zmienić Twoje życie
Opanuj pojęcia nie z tej ziemi
Podłącz się do usług lokalizacyjnych Androida
Zabaw się z bibliotekami Support Libraries
Dawn Griffiths & David Griffiths


Helion
Andrzej Stasiewicz
ANDROID
Podstawy tworzenia aplikacji