Aplikacje typu single Activity

Rafał Zientara

Pobieranie zasobów – przydatne funkcje

- Warto korzystać z funkcji Android Support Library

```
Drawable drawable = AppCompatResources.getDrawable(this, R.drawable.ic_drawable);
Animation animation = AnimationUtils.loadAnimation(this, R.anim.rotate);
MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.dubstep);
int dimensionPixelValue = getResources().getDimensionPixelSize(R.dimen.medium_padding);
int color = ContextCompat.getColor(this, R.color.colorAccent);
String[] stringArray = getResources().getStringArray(R.array.strings);
```

View - przydatne funkcje

- Przez kod aplikacji

```
view.setVisibility(View.GONE);                    //ustawienie widoczności
view.setBackgroundColor(Color.RED);               //kolor tła NIE R.COLOR.IDCOL!
view.setBackground(drawable);                     //drawable na tło
view.setBackgroundResource(R.drawable.ic_drawable); //drawable na tło
view.startAnimation(animation);                   //natychmiostowe uruchomienie animacji
view.setMinimumHeight(dimensionPixelValue);       //minimalna wysokość - zalecane dp
view.setMinimumWidth(dimensionPixelValue);        //minimalna szerokość - zalecane dp
view.setAlpha(floatFromZeroToOne);                //ustawienie przeźroczystości
view.invalidate();                                //wymuszenie odświeżenia View
```

- Przez XML

```
<View
    android:id="@+id/myViewId"
    style="@style/MyStyle"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:layout_margin="8dp"
    android:alpha="120"
    android:background="@color/colorPrimary"
    android:elevation="10dp"
    android:minHeight="20dp"
    android:minWidth="20dp"
    android:onClick="clickView"
    android:padding="8dp"
    android:rotation="30"
    android:scaleX="2"
    android:scaleY="2"
    android:visibility="visible" />
```

View - przydatne funkcje

```
//słuchacz na kliknięcie
view.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

    }
});

//słuchacz na przytrzymanie
view.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        return false;
    }
});

//nasłuchiwanie na akcje ruchów po ekranie
view.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        return false;
    }
});
```

View i ViewGroup - przydatne funkcje

- Dla view (aby nie trzeba było rzutować na ViewGroup)

```
View childView = view.findViewById(R.id.viewObjectId);//szukanie View po id wewnątrz obiektu
View childTag = view.findViewWithTag("tag");          //szukanie elementu wewnątrz po tagu
View focusedView = view.findFocus();                  //znajdź obiekt na którym jest akcja
```

- ViewGroup

```
int childCount = viewGroup.getChildCount();                 //pobieranie ilości dzieci
viewGroup.getChildAt(0);                                    //pobieranie dziecka pod indeksem
viewGroup.addView(childView);                               //dodawanie dziecka
viewGroup.addView(childView, index);                        //dodawanie dziecka pod indeksem
viewGroup.addView(childView, layoutParams);                 //dodawanie dziecka z parametrami układu
```

TextView - przydatne funkcje

```
String text = textView.getText().toString();            //pobieranie tekstu
textView.setText("tekst");                               //ustawianie tekstu
textView.append("tekst na końcu");                       //dodawanie tekstu na koniec teksu
textView.setEms(marksCount);                             //szerokość widoku po liczbie znaków
textView.setError("Wypełnij pole!");                     //wyświetl treść błędu
textView.setHint(R.string.hintText);                     //wyświetl tekst z poradą
textView.setAllCaps(true);                               //wyświetl tekst jako CAPS LOCK
textView.setEnabled(true);                               //tekst jest interaktywny lub nie
textView.setLines(linesCount);                           //ilość linii jako integer

//nasłuchiwanie zmian tekstu
textView.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
    }
    @Override
    public void afterTextChanged(Editable s) {
    }
});
```

ImageView - przydatne funkcje

- W kodzie

```
imageView.setAdjustViewBounds(true);                    //dopasowanie się do rozmiaru View
imageView.setImageAlpha(120);                           //ustawianie przeźroczystości obrazka
imageView.setImageDrawable(drawable);                   //ustawianie drawable
imageView.setImageBitmap(bitmap);                       //ustawianie bitmapy
imageView.setImageResource(R.drawable.image);           //ustawianie drawable na podstawie id zasobu
imageView.setScaleType(ImageView.ScaleType.FIT_XY); //różny style skalowania obrazu
```

- W XML

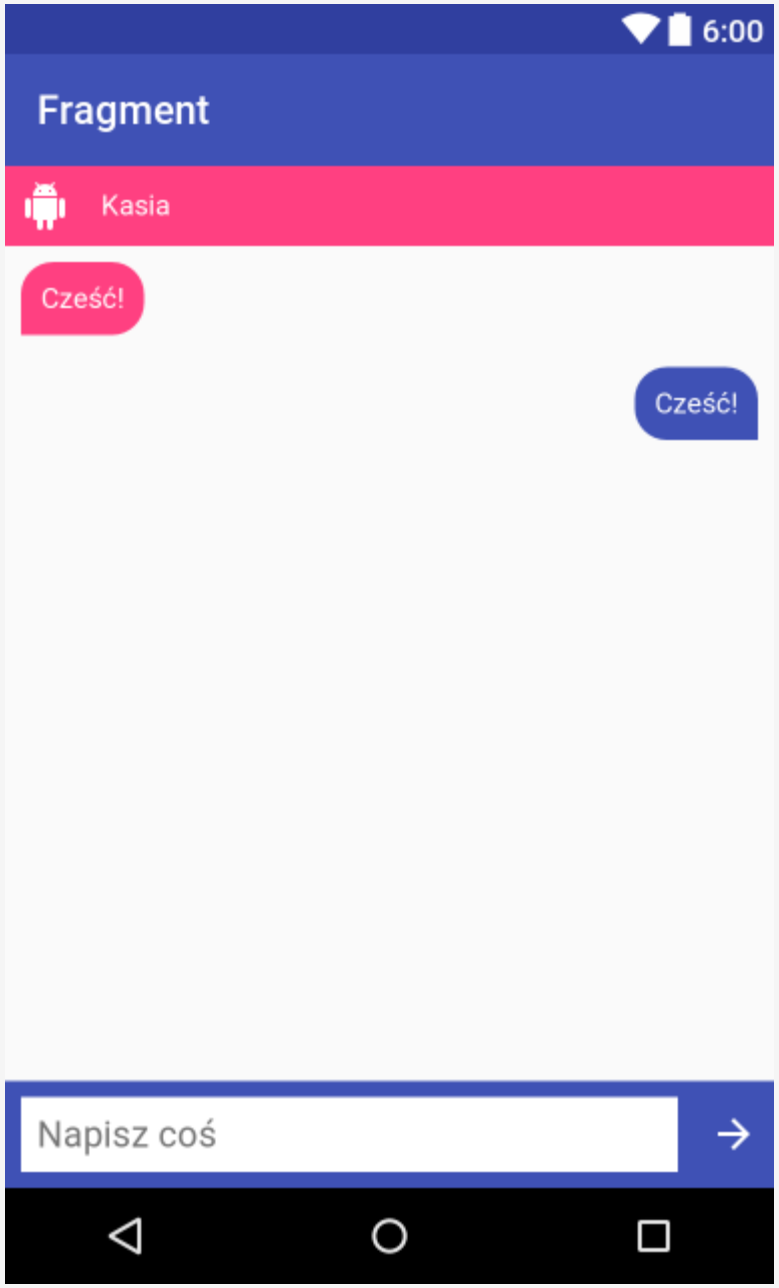
```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:adjustViewBounds="true"
    android:scaleType="fitXY"
    android:src="@drawable/image"
    app:srcCompat="@drawable/image" />
```

Zadanie 1

Stwórz widok według konceptu

Dodatkowo:

- Dodawaj po kliknięciu strzałki wpisany tekst do czatu



Zadanie 2

Kamień Papier Nożyce (projekt MVC)

- Stwórz grę gdzie na głównym ekranie będą 3 przyciski do wyboru akcji u dołu ekranu
- Na górze dwie duże ikony pokazujące co wybraliśmy my, a co komputer
- Na górze wyświetlaj wynik kto zdobył ile punktów
- Gdy wybierzemy akcję wyświetli się wybór komputera i okno dialogowe (AlertDialog/DialogFragment) z informacją o wygranej/przegranej
- Logikę gry umieść w klasie Model

W oknie dialogowym do wybrania opcje:

- Jeszcze raz – widok zostaje zresetowany
- Już mam dość