Pokračovanie

Layout, View, Intent
List, Canvas, Menu

Peter Borovanský
KAI, I-18

borovan 'at' ii.fmph.uniba.sk

Hitparáda

(Hall of Fame)

Kalkulačka, 12 riešení

DominikaK, DenisČ, ŠimonB, ZuzkaH, LindaJ, MonikaV

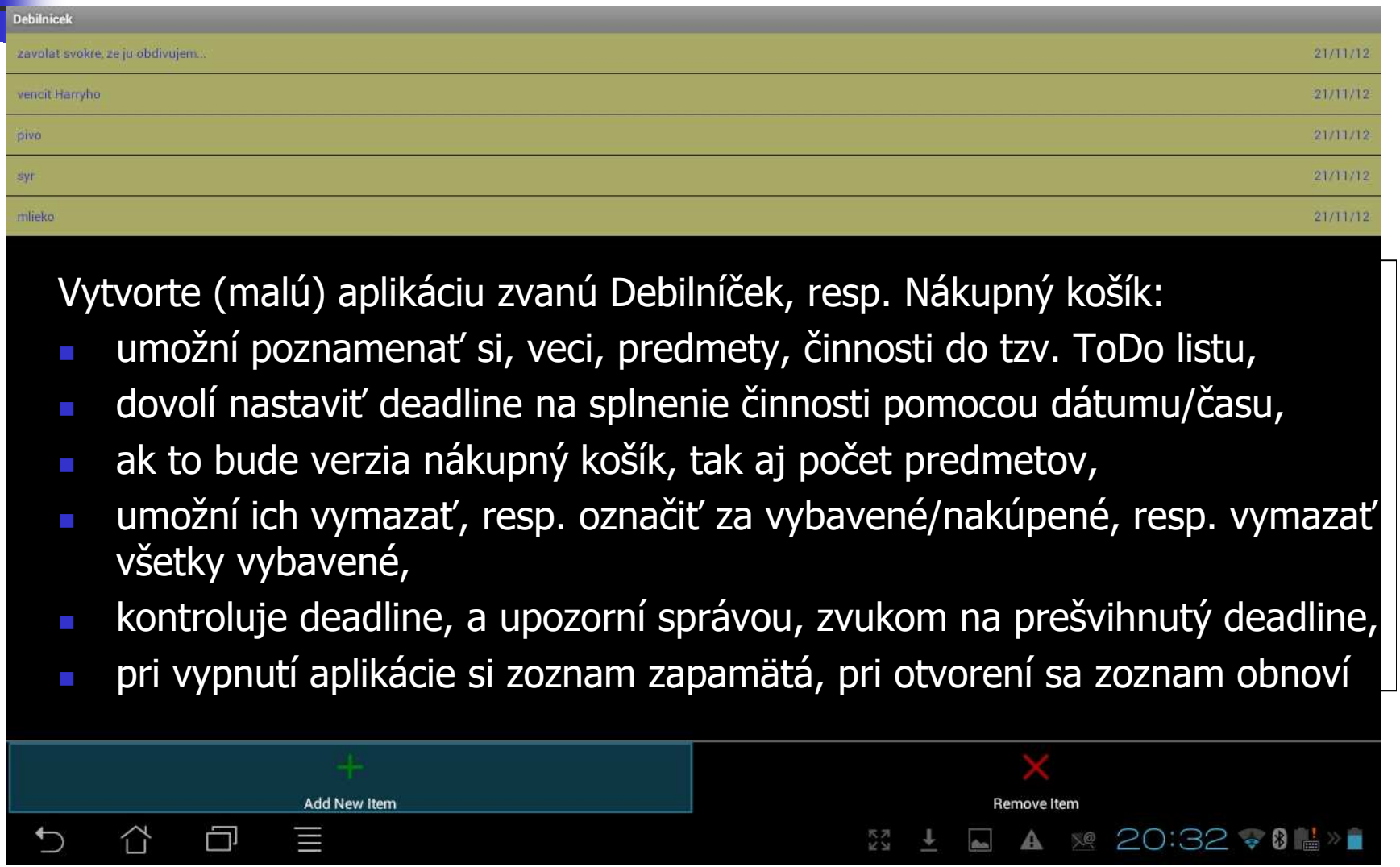
```
java:
Integer.toString(15,7) = "21"
Integer.parseInt(Integer.toString(15,7))
   = 21

Integer.parseInt("21",7) = 15
Integer.parseInt(""+21,7) = 15
```

```
kotlin:
println(15.toString(7))
21
```

```
println("21".toInt(7))
15
```

Domáca úloha 2

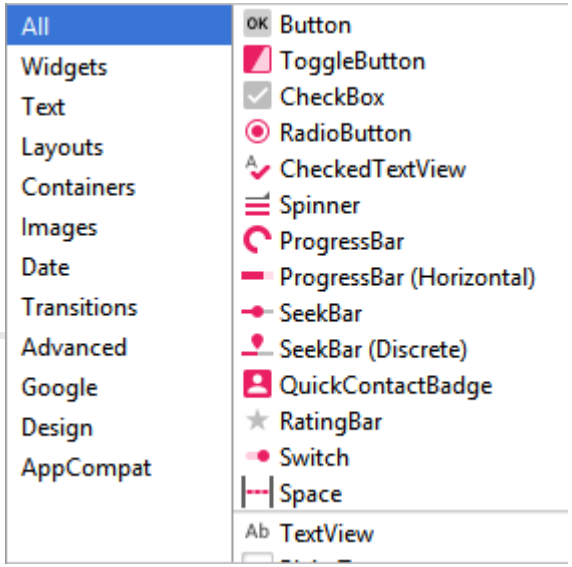
(deadline do 10.nov)



Vytvorte (malú) aplikáciu zvanú Debilníček, resp. Nákupný košík:

- umožní poznamenať si, veci, predmety, činnosti do tzv. ToDo listu,
- dovolí nastaviť deadline na splnenie činnosti pomocou dátumu/času,
- ak to bude verzia nákupný košík, tak aj počet predmetov,
- umožní ich vymazať, resp. označiť za vybavené/nakúpené, resp. vymazať všetky vybavené,
- kontroluje deadline, a upozorní správou, zvukom na prešvihnutý deadline,
- pri vypnutí aplikácie si zoznam zapamätá, pri otvorení sa zoznam obnoví

GUI komponenty



Layout

- LinearLayout (Verical/Horizontal)
- RelativeLayout, ConstraintLayout

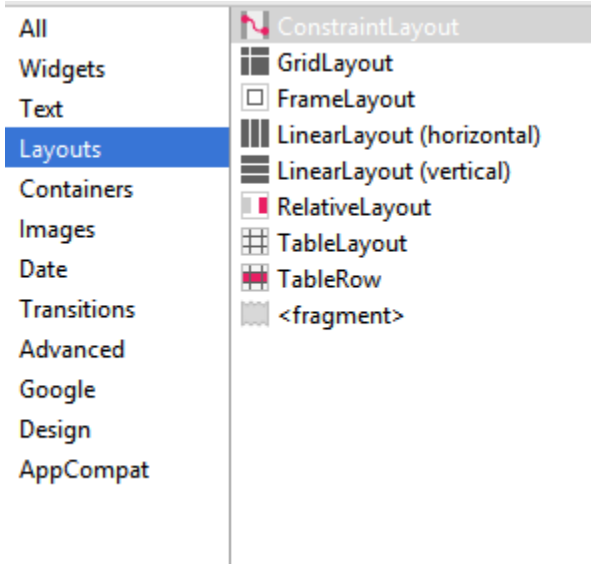
View, ViewGroup

- všetky viditeľné komponenty (widgets)

Activity - analógia Screenu (MIT), resp. Form/Frame

najznámejšie podtriedy

- ListActivity pre ListView, zobrazenie zoznamu
- MapActivity pre MapView (zobrazenie mapy)



Fragment (>= API level 11)

- reusable UI components

Layouts
(match_parent, wrap_content)

- FrameLayout – objeky umiestni v ľavom hornom rohu
- LinearLayout – horizontálny/vertikálny
- RelativeLayout – dovolí umiestniť objekty relatívne k pozíciám iných objektov
- ConstraintLayout (support library, API 9, od Android Studio 2.2)
- GridLayout (od API Level 14)

```
<FrameLayout
        android:id="@+id/FrameLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        <ImageView
          android:id="@+id/imageView1"
          android:layout_width="match_parent" --roztiahni podľa
          android:layout_height="match_parent" -- rodičovského
          android:src="@drawable/ic_launcher" />

</FrameLayout>
```

Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

LinearLayout


```
<LinearLayout
    android:orientation="vertical"
    <LinearLayout
       android:orientation="horizontal"
        <TextView
               android:id="@+id/lb1"
               android:text="@string/login"/>
       <EditText
               android:id="@+id/logintv"
               android:layout_width="match_parent" --roztiahni
               android:layout_height="wrap_content"-na výšku fontu
               android:inputType="textEmailAddress" /> -- filter
    </LinearLayout>
  ... podobne  pre password
```

Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

LinearLayout

(weight, gravity, align with the base line)


```
<LinearLayout   ... Pokračovanie
        <LinearLayout
        android:orientation="horizontal"
        <Button
                android:id="@+id/logBtn"
                android:layout_weight="50"
                android:text="@string/Login"/>
        <Space
                android:layout_weight="50" />
        <Button
                android:id="@+id/forgetPass"
                android:layout_weight="50"
                android:text="@string/forget" />
        </LinearLayout>
</LinearLayout>
```

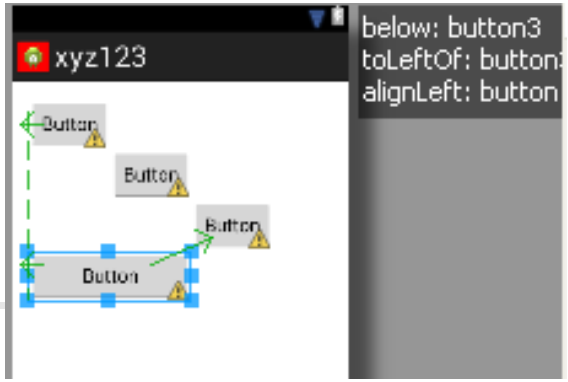
Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

GridLayout


```
<GridLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:rowCount="4">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        android:id="@+id/button1"
        android:layout_row="0"
        android:layout_column="0" />

    <Button ...
        android:layout_row="0"
        android:layout_column="1" />
```

Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

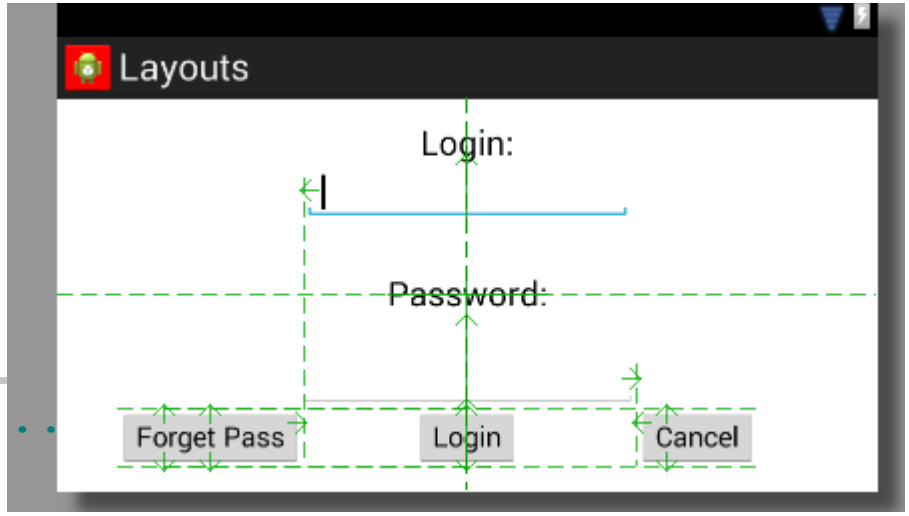
RelativeLayout




```
<RelativeLayout
    <Button
        android:id="@+id/button1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"/>
    <Button
        android:id="@+id/button2"
        android:layout_below="@+id/button1"
        android:layout_toRightOf="@+id/button1"/>
... <Button
        android:id="@+id/button4"
        android:layout_alignLeft="@+id/button1"
        android:layout_below="@+id/button3"
        android:layout_toLeftOf="@+id/button3" />
</RelativeLayout>
```

Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

RelativeLayout



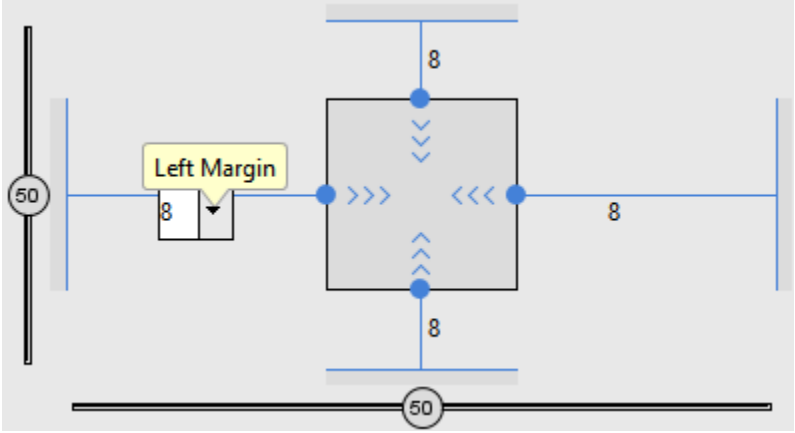

```
<RelativeLayout> ... skrátené...
    <EditText
                android:id="@+id/passwdtv"
                android:layout_below="@+id/pass"
                android:layout_centerHorizontal="true"/>
     <Button
                android:id="@+id/loginBtn"
                android:layout_below="@+id/passwdtv"
                android:text="@string/Login" />
     <Button
                android:id="@+id/forgetBtn"
                android:layout_alignBottom="@+id/loginBtn"
                android:layout_alignTop="@+id/loginBtn"
                android:layout_toLeftOf="@+id/passwdtv"
                android:text="@string/forget" />
</RelativeLayout>
```

originál nájdete v Layouts2.zip

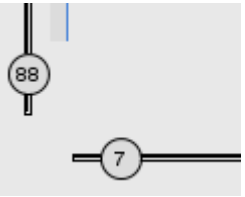
Constraint Layout

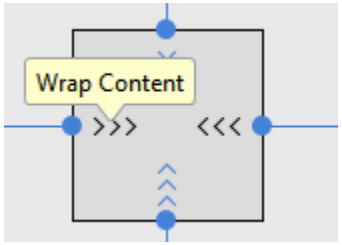
Umožňuje nastaviť väzby

- relatívnu pozíciu
- spoločná baseline pre text
- okraje
- wrap/match content/fixná veľkosť
- vychýlenie (bias)



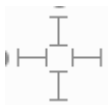
https://developer.android.com/reference/android/support/constraint/ConstraintLayout.html
https://www.youtube.com/watch?v=z53Ed0ddxgM



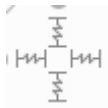


Kód na slajde je zjednodušený, originál nájdete v Layouts2.zip

Constraint Layout

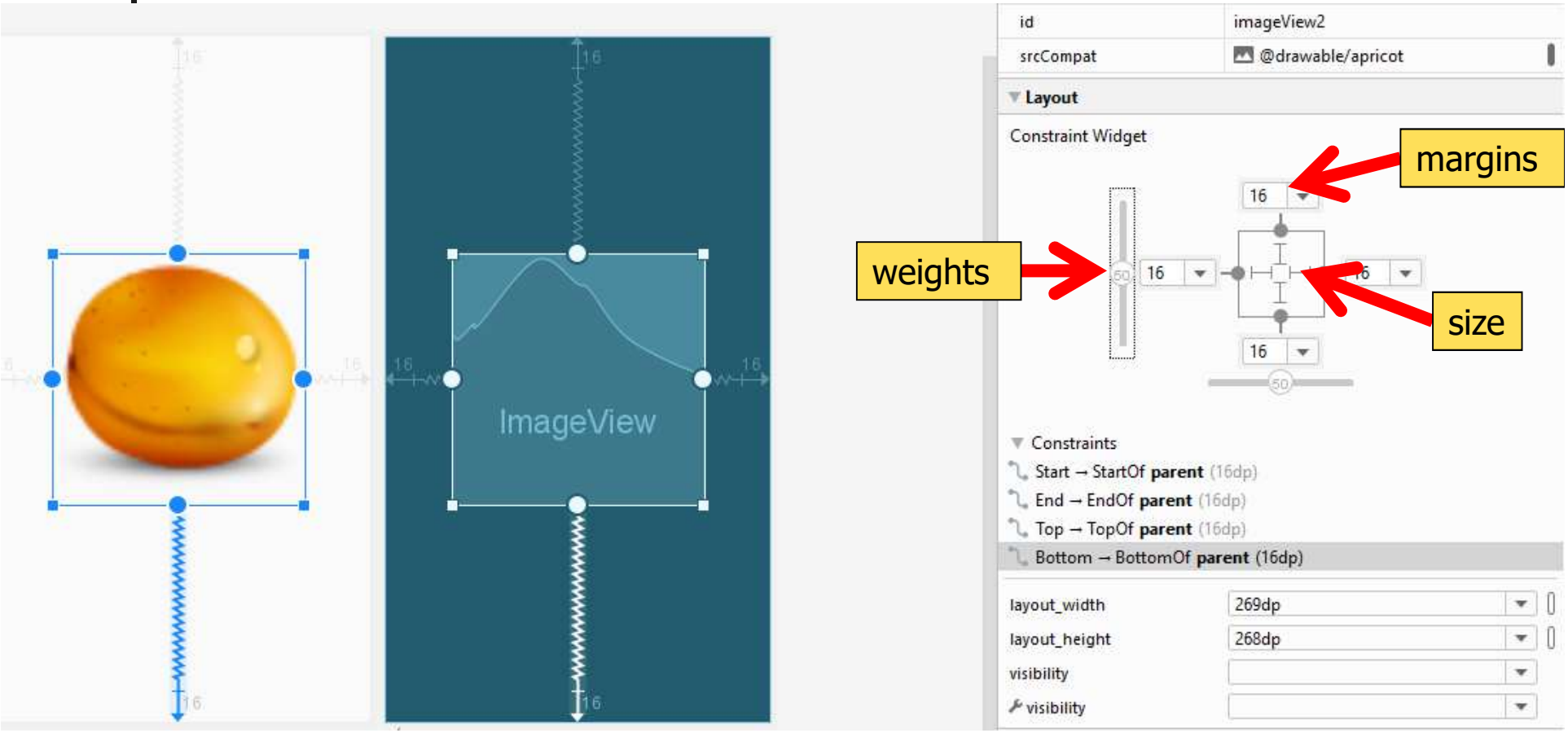


fixed size in dp ☹



match parent

wrap content

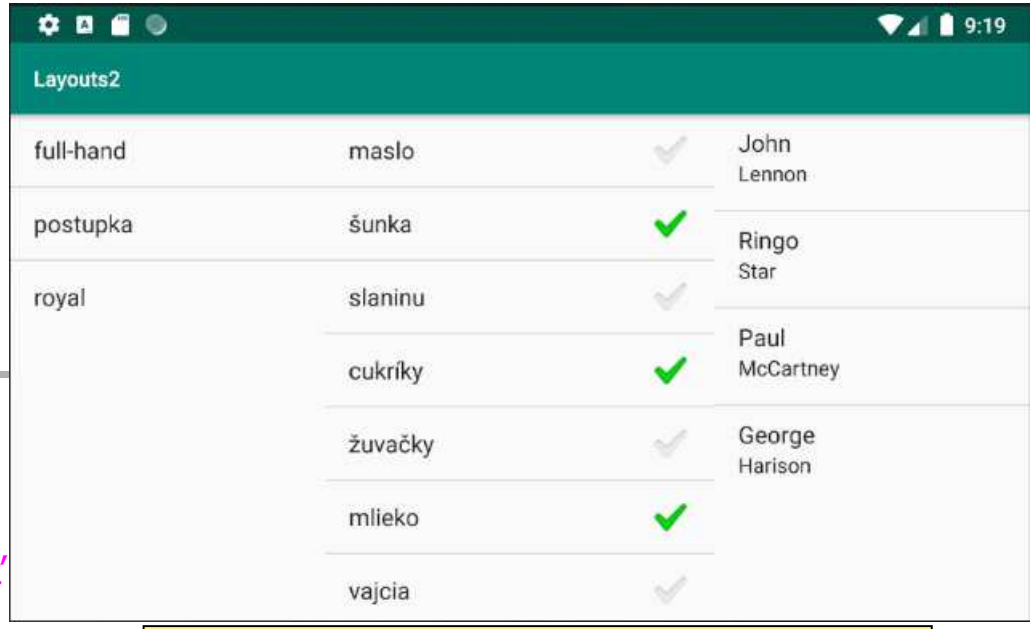


ListView

(variabilita)

ListView a ListActivity zobrazujú zoznam položiek a môžu mať

- preddefinovaný štýl
  - môžu/nemusia sa nám páčiť
  - ale sú ready to use
- user defined
  - narobíme sa pri ich definícii



Rôzne inštancie ListView

simple_list_item_1,
simple_list_item_activated_1
simple_list_item_checked
simple_list_item_2
....

Odchytávanie udalostí v ListView

```
com.example.layouts2 D/ZOZNAM: beatles click: 2:{krstne=Paul, priezv=McCartney}
com.example.layouts2 D/ZOZNAM: beatles click: 1:{krstne=Ringo, priezv=Star}
com.example.layouts2 D/ZOZNAM: beatles click: 3:{krstne=George, priezv=Harison}
com.example.layouts2 D/ZOZNAM: check click: 3:cukríky
com.example.layouts2 D/ZOZNAM: check click: 4:žuvačky
com.example.layouts2 D/ZOZNAM: item click: 1:postupka
com.example.layouts2 D/ZOZNAM: item click: 2:royal
com.example.layouts2 D/ZOZNAM: item click: 0:full-hand
com.example.layouts2 D/ZOZNAM: check click: 2:slaninu
```

Project: Layouts2.zip

ListView

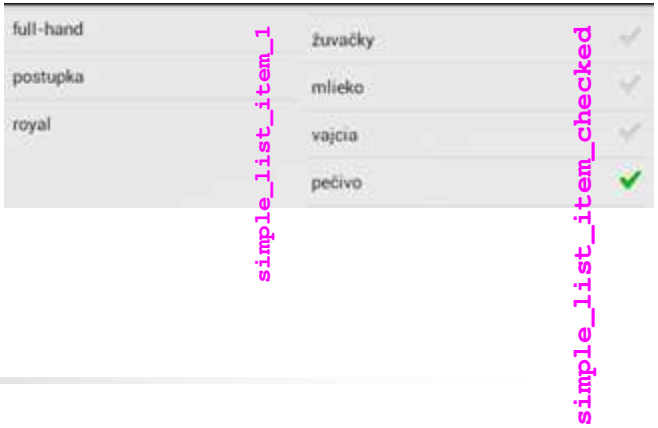
(simple_list_item_1)


```
// poker - simple_list_item1 view
listView1.adapter = ArrayAdapter<String>(
    this,
    android.R.layout.simple_list_item_1, // jednoriadkový
       // simple_list_item_activated_1
    resources.getStringArray(R.array.poker) // hodnoty
)

// listView1.choiceMode = ListView.CHOICE_MODE_MULTIPLE


listView1.setOnItemClickListener {
   adapterView, view, index, l -> // View.OnItemClickListener
      val hodnota = adapterView.getItemAtPosition(index)
      Log.d(TAG, "item click: $index:$hodnota")
}
```

Project: Layouts2.zip

ListView

(simple_list_item_checked)

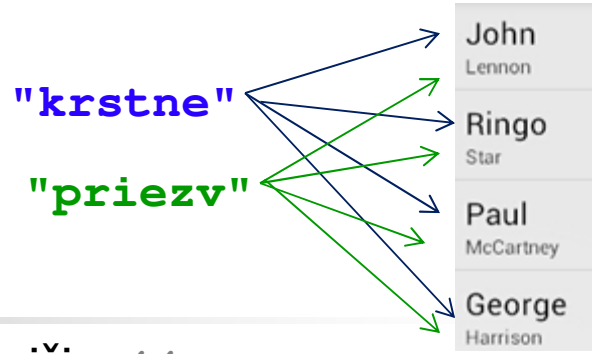



```
// nákup - checked box list view
listView2.adapter = ArrayAdapter<String>(
    this,
    android.R.layout.simple_list_item_checked, //2riadkový
    resources.getStringArray(R.array.nakup)
)

listView2.setOnItemClickListener {
    adapterView, view, index, l ->
       val hodnota = adapterView.getItemAtPosition(index)
       (view as CheckedTextView).toggle() // prekresli
       Log.d(TAG, "check click: $index:$hodnota")
}
```

Project: Layouts2.zip

ListView 2

(simple_list_item_2)



Naplniť iný, napr. dvojriadkový ListView je náročnejšie

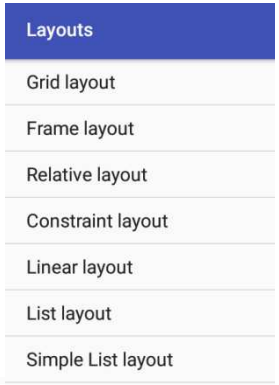
```
//beatles list view
val pairs = listOf( // hodnoty sú zoznam máp kľúč->hodnota
    mapOf("krstne" to "John","priezv" to "Lennon"),mapOf("krstne" to "Ringo","priezv" to "Star"),
    mapOf("krstne" to "Paul","priezv" to "McCartney"),mapOf("krstne" to "George","priezv" to "Harison")
)
listView3.adapter = SimpleAdapter(this,
     pairs,                                  // hodnoty
     android.R.layout.simple_list_item_2, // format ListView
     arrayOf("krstne", "priezv"),            // zoznam kľúčov
     arrayOf(android.R.id.text1,android.R.id.text2) // riadky
        .toIntArray()
   )
   listView3.setOnItemClickListener {
        adapterView, view, index, l ->
       val hodnota = adapterView.getItemAtPosition(index)
       Log.d(TAG, "beatles click: $index:$hodnota:"+
                "${(hodnota as Map<String, String>)["krstne"]
}
```

Rôzne preddefinované ListView

(prehľad)

zdroj: https://developer.xamarin.com/guides/android/user_interface/working_with_listviews_and_adapters/part_3_-_customizing_a_listview's_appearance/

Intent
(filter)



Pohľad do AndroidManifest: intent-filter hovorí, na aký intent aktivita reaguje

```
<activity android:name=".MainActivity">
   <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
   </intent-filter>
</activity>
```

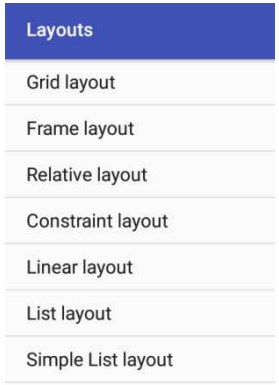
Spustí sa ako prvá

```
<activity android:name=".ListActivity">
   <intent-filter>
      <category android:name="android.intent.category.DEFAULT" />
   </intent-filter>
</activity>
```

nespustí sa

Project: Layouts2.zip

Intent
(startActivity)




```
listViewID.adapter = ArrayAdapter<String>(
    this, android.R.layout.simple_list_item_1,
    arrayListOf("Grid layout","Frame layout","Relative layout",
          "Constraint layout",)) // dáta nepatria do kódu, ale .xml
    resources.getStringArray(R.array.activities)
```

```
listViewID.setOnItemClickListener {
    adapterView, view, index, l ->
    Log.d("LISTPICK",
         "click: $index:${adapterView.getItemAtPosition(index)}")
    if (index < klasy.size)
       startActivity(Intent(this@MainActivity,klasy[index]))
}
```

```
private val klasy = arrayOf(
    GridLayoutActivity::class.java,
    FrameLayoutActivity::class.java,
    ... .
    MainActivity::class.java
)
```

Project: Layouts2.zip

List project

V ďalšom uvidíme sériu rôznych nezávislých aktivít, ktoré ilustrujú:

- intro_activity
  - logo, intent, CountDown/Timer, MediaPlayer
- email_activity
  - listView, intent.putExtra, startActivityForResult, Toast
- canvas_activity
  - canvas/view – Draw, MultiTouch, onTouch, Option & Context Menu
- pisky_activity
  - piškvorky, začiatok aj koniec jednoduchej hry
- login_activity
  - ukladanie informácie pomocou SharedPreferences

Project: List.zip

Intent - filter

- CATEGORY_BROWSABLE - ovláda web browser
- CATEGORY_LAUNCHER - ovláda spúšťač aplikácie
- *android.intent.action.MAIN – vstupný bod programu*

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

- CATEGORY_DEFAULT – startActivity/startActivityForResults

```
<intent-filter>
    <action android:name="com.example.actilist.CanvasActivity" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

spustenie:

```
startActivity(
    Intent(this@IntroActivity, MainActivity::class.java))
```

ak máme:

```
class MainActivity : AppCompatActivity() {
```

Project: List.zip

Reflexivita

Aby sme nemuseli mať konštantu ako pole všetkých tried, trieda sa dá vyrobiť z mena triedy pomocou reflexívneho volania **`Class.forName`**

```
class MainActivity : AppCompatActivity() {
...
listView1.setOnItemClickListener {
  adapterView, view, index, l ->
    val hodnota = adapterView.getItemAtPosition(index)
    Log.d(TAG, "list item click: $index:$hodnota")
    val klasa = Class.forName("com.example.list.$hodnota")
                .kotlin
    //val intent = Intent(this, IntroActivity::class.java)
    Log.d(TAG, "class name: ${klasa.qualifiedName}")
    val intent = Intent(klasa.qualifiedName)
    startActivity(intent)
}
```

Project: List.zip

IntroActivity

(Intent, timer)

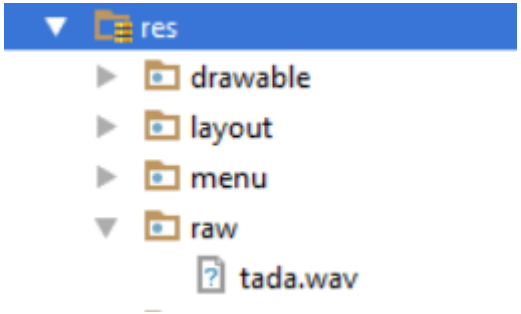
IntroActivity – CountDownTimer odpočítavajúci čas pre úvodné logo+.mp3

```
override fun onCreate(savedInstanceState: Bundle?) {
   super.onCreate(savedInstanceState)
   setContentView(R.layout.activity_intro)
   object: CountDownTimer(4000,1000) {
      override fun onTick(millisUntilFinished: Long) {}
      override fun onFinish() {
         Log.d(TAG, "go back to mainActivity")
         startActivity(
            Intent(this@IntroActivity, MainActivity::class.java)
         )
      }
   }.start()
```

Project: List.zip

MediaPlayer

(lokálne – adresár raw)



tada.mp3 [.wav] uložíme do project/res/raw
... a bude zakompilovaná do apky, a zazipovaná do zipky ☺

```
lateinit var mp : MediaPlayer          // globálna premenná
// ak je muzička lokálna, v res/raw/tada.wav
mp = MediaPlayer.create(this, R.raw.tada)
mp.isLooping = false
mp.start()
```

```
override fun onPause() {       // ak je IntroActivity pauzovaná, keď
   super.onPause() // odštartujeme com.example.actilist.MainActivity
   mp.release()    // uvoľníme MediaPlayer objekt, mp
   finish()         // akonáhle sa rozbehne MainActivy,
                   // IntroActivity  zanikne
                   // to rieši aj navigáciu, problém s back buttonom
}
```

Project: List.zip

MediaPlayer
(onPreparedListener)

iná možnosť, tada.mp3 je prístupná niekde na sieti, dotiahneme ju a zahráme

```
    // problém:apka musí deklarovať, že chce prístup na internet
lateinit var mp : MediaPlayer
try { // ak je muzička na webe, jej dotiahnutie może niečo trvať
    val uri = Uri.parse("http://dai.fmph.uniba.sk/courses/VMA/wave.mp3")
    mp.setAudioStreamType(AudioManager.STREAM_MUSIC)

    mp.setOnPreparedListener { mp.start() }

    mp.setDataSource(getApplicationContext(), uri)
    mp.prepare()     // tu sa spustí doťahovanie súboru
} catch (e:IOException) {
    e.printStackTrace()
    Toast.makeText(this, "file error", Toast.LENGTH_SHORT).show()
}
```

do AndroidManifest.xml

treba deklarovať povolenie aplikácie prístupu na internet

```
<uses-permission android:name="android.permission.INTERNET" />
```

http://dai.fmph.uniba.sk/courses/VMA/wave.mp3

Project: List.zip

MediaPlayer

(na SD-karte, resp. v internej pamäti)

Môžeme sa skúšať triafať do správnej cesty muziky, obrázku, či súboru:

```
mp.setDataSource("/mnt/sdcard/Music/tada.wav")
mp.setDataSource("/mnt/sdcard/Music/wave.mp3")
mp.setDataSource("/storage/sdcard0/Music/wave.mp3")
mp.setDataSource("/Removable/SD/Music/wave.mp3")

// ale správna cesta k prístupu k Music je cez root external storage
val filePath = Environment.getExternalStorageDirectory().toString()
                                + "/Music/tada.wav"
Log.d(TAG, filePath)   // vždy si zalogujte cestu,
                       // aby ste vedeli, kde súbor hľadá
mp = MediaPlayer()
mp.setDataSource(filePath) // hneď viete, prečo to nehrá..
mp.setOnPreparedListener { mp.start() }
mp.prepare()
```

Project: List.zip

EmailActivity

(data do intentu, startActivityForResult s callbackom)

```
val emailString = edtEmail.text.toString() // dáta z formulára
val subjectString = edtSubject.text.toString()
val bodyString = edtBody.text.toString()

Toast.makeText(this@EmailActivity, "posielam mail",
               Toast.LENGTH_LONG).show()

val intent = Intent(android.content.Intent.ACTION_SEND) // SEND

intent.type = "text/plain"
intent.putExtra(android.content.Intent.EXTRA_SUBJECT,subjectString)
intent.putExtra(android.content.Intent.EXTRA_EMAIL,
                arrayOf(emailString) ) // pole adresátov
intent.putExtra(android.content.Intent.EXTRA_TEXT, bodyString)

// startActivity(intent);
startActivityForResult(intent, REQUEST_SEND_EMAIL)

private val REQUEST_SEND_EMAIL = 777
```

Project: List.zip

EmailActivity

(onActivityResult = callback)

```
private val REQUEST_SEND_EMAIL = 777

override fun onActivityResult( // CALLBACK pre startActivityForResult
        requestCode: Int,
        resultCode: Int,
        data: Intent?) {

    // Check which request we're responding to
    if (requestCode == REQUEST_SEND_EMAIL) {
        Toast.makeText(
            this@EmailActivity, "email poslany, alebo aj nie",
            Toast.LENGTH_SHORT
        ).show();
    }
}
```

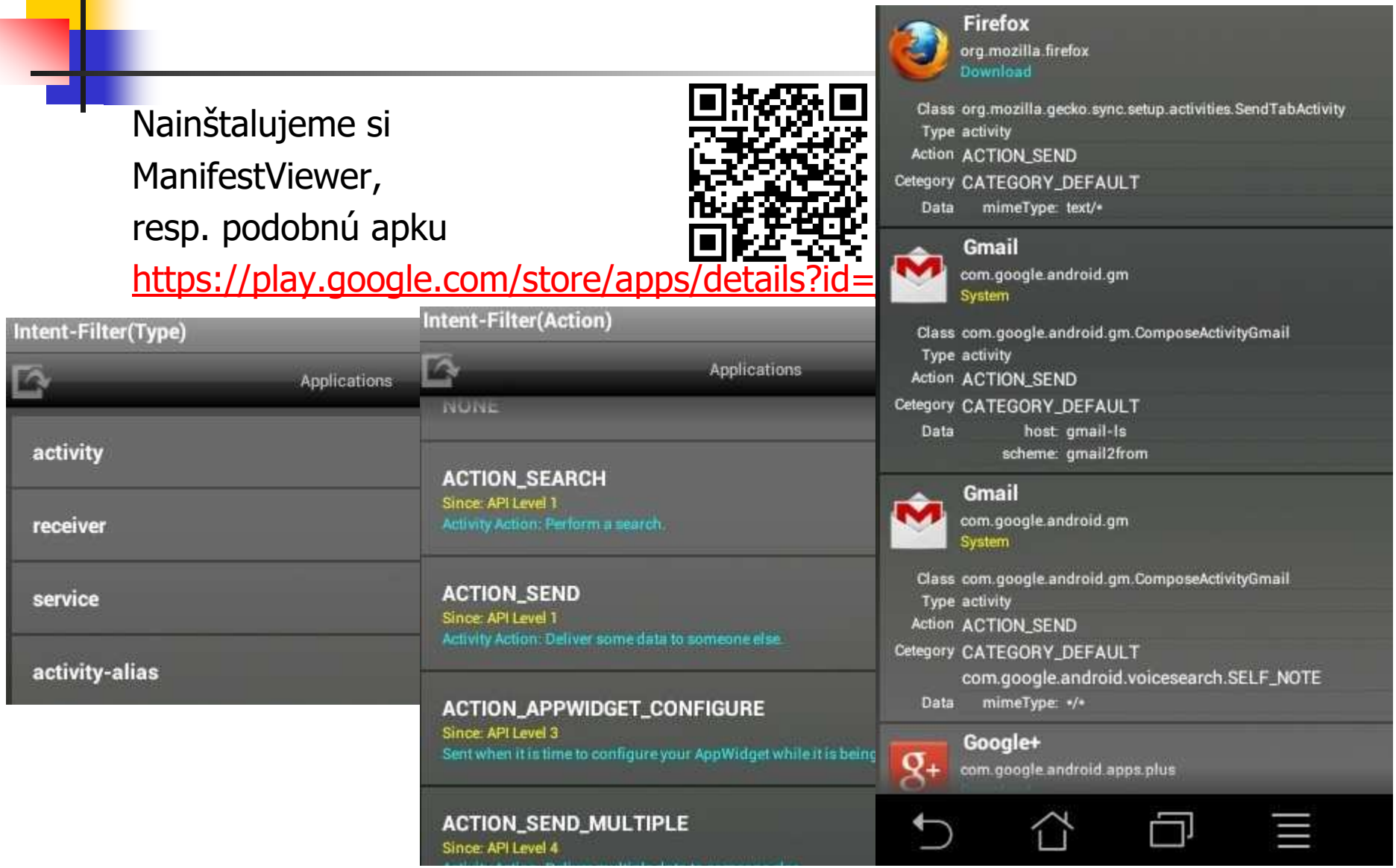
https://developer.android.com/reference/android/content/Intent.html#ACTION_SEND

Project: List.zip

Kto všetko chytá intent ?

`android.content.Intent.`*`ACTION_SEND`*

Nainštalujeme si
ManifestViewer,
resp. podobnú apku
https://play.google.com/store/apps/details?id=



PhotoActivity

(data z intentu)

Princíp intent-startActivityForResult spolu s onActivityResult ešte raz:

```
val takePictureIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE);
if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
   Toast.makeText(this@PhotoActivity,
           "smile ... taking picture", Toast.LENGTH_LONG).show();

   startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
} else {
   Toast.makeText(this@PhotoActivity,
           "sorry ... no picture", Toast.LENGTH_LONG).show();
}

private val REQUEST_IMAGE_CAPTURE = 690
```

https://developer.android.com/reference/android/provider/MediaStore.html#ACTION_IMAGE_CAPTURE

Project: List.zip

PhotoActivity
(data z intentu)

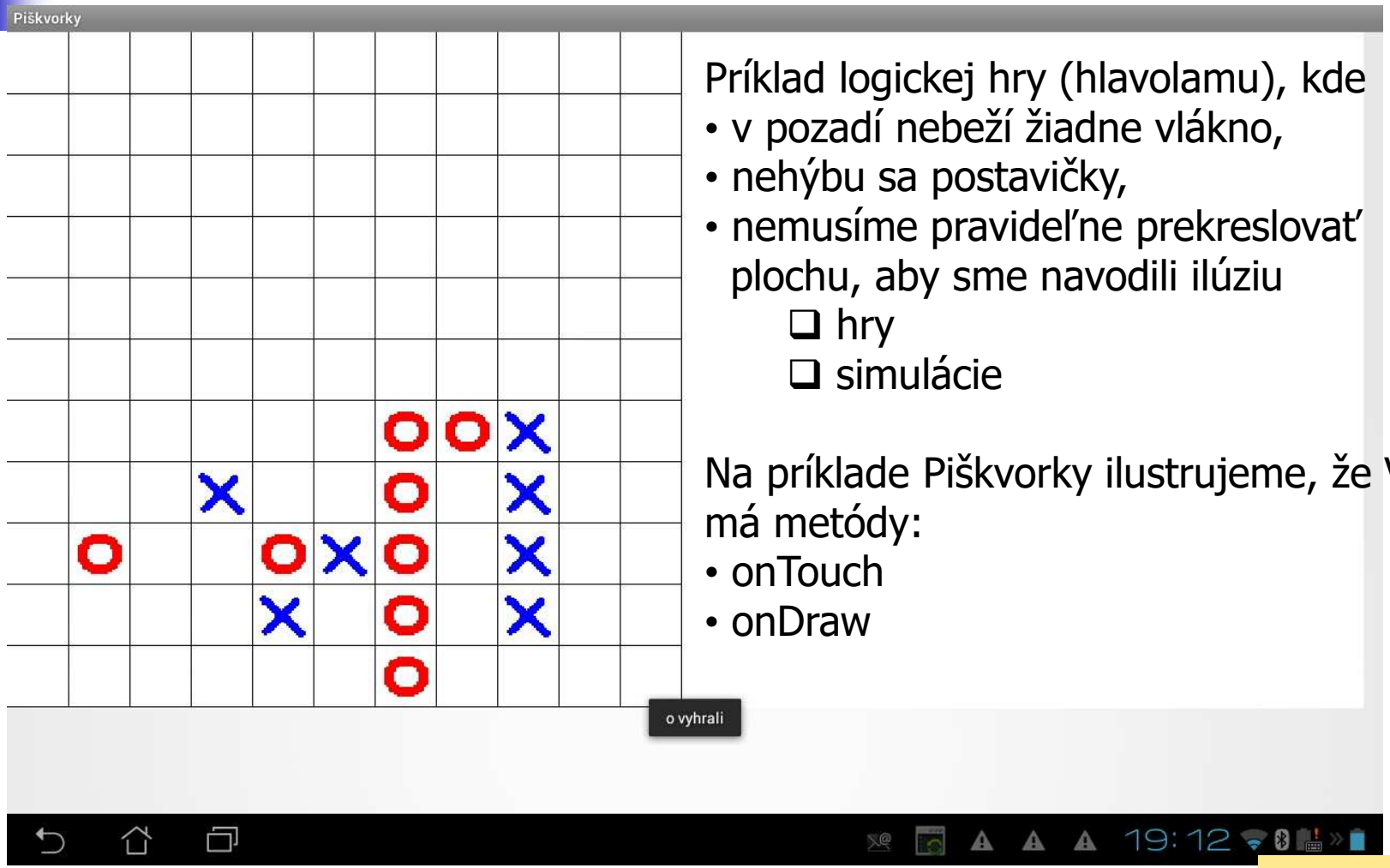
V callback onActivityResult získavame z indentu data/odfotený obrázok:

```
private val REQUEST_IMAGE_CAPTURE = 690

override fun onActivityResult(requestCode: Int,
   resultCode: Int, data: Intent?) {
   // Check which request we're responding to
   if (requestCode == REQUEST_IMAGE_CAPTURE &&
      resultCode == RESULT_OK) {
      Toast.makeText(this@PhotoActivity, "thanks ...",
                     Toast.LENGTH_LONG).show()
      val extras = data?.getExtras()
      val imageBitmap = extras?.get("data") as Bitmap
      pictureImageView.setImageBitmap(imageBitmap)
   }
```

Project: List.zip

Piškvorky

(logická hra v canvase)



Príklad logickej hry (hlavolamu), kde

- v pozadí nebeží žiadne vlákno,
- nehýbu sa postavičky,
- nemusíme pravidelne prekreslovať plochu, aby sme navodili ilúziu
  - ❑ hry
  - ❑ simulácie

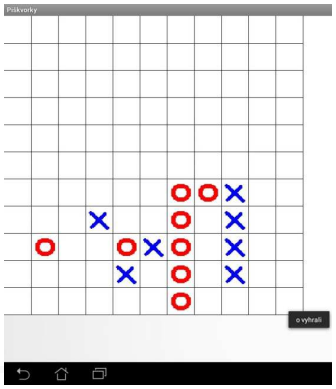
Na príklade Piškvorky ilustrujeme, že View má metódy:

- onTouch
- onDraw

Project:List.zip

onTouch vo View

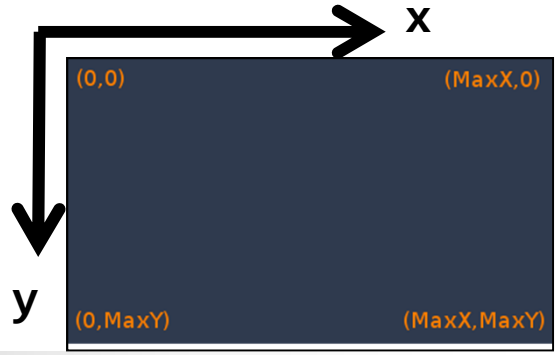
(onTouchEvent)



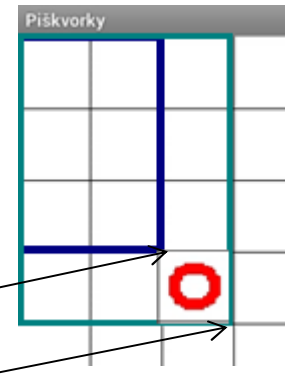
```
class PiskyView(context: Context, attrs: AttributeSet) :
   View(context) {   // Piskvorky sú View
                     // načítanie bitmapy obrázkov, postavičiek
    o_img = resources.getDrawable(R.drawable.o).toBitmap()
    x_img = resources.getDrawable(R.drawable.x).toBitmap()

override  fun onTouchEvent(e:MotionEvent): Boolean {
   if (e.action == MotionEvent.ACTION_DOWN) {
      val iX = (e.x / cellSize).toInt()              // transformácia
      val iY = (e.y / cellSize).toInt()              // pixlov na bunku
      if (iX >= SIZE || iY >= SIZE) return true      // mimo hraciu dosku
       if (playGround[iY][iX] == -1) {               // voľné políčko ?
          playGround[iY][iX] = onTurn                // polož značku hráča
          onTurn = 1 - onTurn                        // na ťahu, a ide súper
          invalidate()        // toto nakoniec prekreslí view
           val winner = check(iX, iY)                // vyhodnotenie víťazov...
           if (winner != -1)
              Toast.makeText(getContext(),"x vyhrali",Toast.LENGTH_LONG).show(
            } else
```

Project:List.zip

onDraw vo View




```
override protected fun onDraw(canvas: Canvas) {//  paint()
    canvas.drawColor(Color.WHITE)
    val p = Paint()
    p.setColor(Color.BLACK)
    p.setStrokeWidth(1F)
    for (i in 1..SIZE) {
      canvas.drawLine(i*cellSize, 0F, i*cellSize, minSize, p)
      canvas.drawLine(0F,i*cellSize, minSize, i*cellSize, p)
    }

    for (y in 0 until SIZE) {
       for (x in 0 until SIZE) {
              ...
          canvas.drawBitmap(o_img,srcRect,
                                  destRect,
                                  p);
```

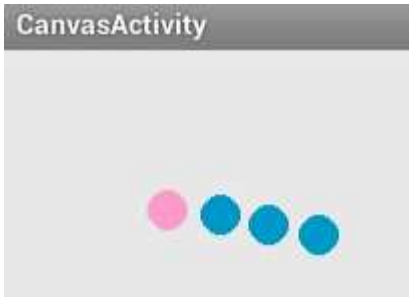


Project:List.zip

Vlákno (Thread) vo View

(dynamická hra v canvase)

```
class CanvasView(context: Context, attrs: AttributeSet) :
    View(context), View.OnTouchListener, View.OnKeyListener {
    var touchX = 100f; var touchY = 100f
    var ballX = 200f;  var ballY = 200f

init {
    setOnTouchListener(this) setOnKeyListener(this)
    val th = object : Thread() {
        override fun run() {
            while (!stopped) {
                if (!paused) {
                    ballX += (touchX-ballX)/touches/50
                    ballY += (touchY-ballY)/touches/50
                    touchX = (ballX+50*touchX[i])/51
                    touchY = (ballY+50*touchY[i])/51
                    try {
                        Thread.sleep(100)
                        postInvalidate()
                    } catch (e: InterruptedException) {
    th.start()
```



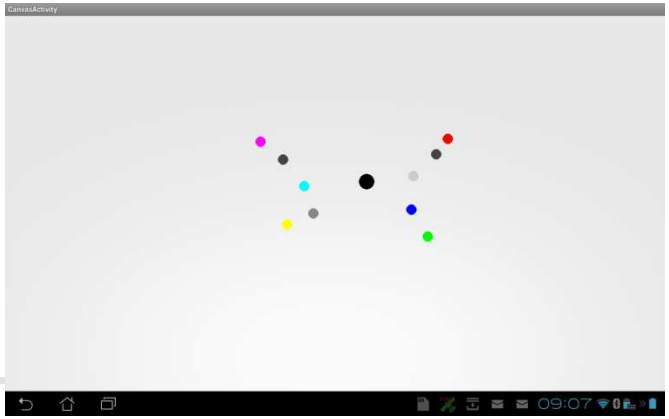
Project:List.zip

onDraw, onTouch vo View

```
@Override
protected void onDraw(Canvas canvas) { // paint z Appletov
    super.onDraw(canvas);
    if (canvas != null) {
        Paint p = new Paint();            // kreslenie guličiek
        p.setColor(getResources().getColor(R.color.red));
        canvas.drawCircle(touchX, touchY, 10, p);
        p.setColor(getResources().getColor(R.color.blue));
        canvas.drawCircle(ballX, ballY, 10, p);
    } else
        Log.d("Canvas", "null");
}
public boolean onTouch(View v, MotionEvent event) {
    touchX = event.getX();                // netestujeme typ eventu
    touchY = event.getY();                // zoberieme len X,Y súradnice
    return true;
}
```

Project:List.zip

MultiTouch



```
override fun onTouch(v: View, event: MotionEvent): Boolean {
    Log.d("Canvas", "counts:" + event.pointerCount)
    val maskedAction = event.actionMasked
    if (maskedAction == MotionEvent.ACTION_DOWN ||
        maskedAction == MotionEvent.ACTION_POINTER_DOWN) {
        touches = event.pointerCount
        for (i in 0 until event.pointerCount) {
            Log.d("Canvas", "X:" + event.getX(i))
            Log.d("Canvas", "Y:" + event.getY(i))
            touchX[i] = event.getX(i)
            touchY[i] = event.getY(i)
        }
        return true
```

Žiadne dva prsty sa nedotknú naraz

Project:List.zip

onKey vo View

```
override fun onKey(arg0: View, arg1: Int, arg2: KeyEvent):
                Boolean {
   val rnd = Random()
   when (arg1) {
      KeyEvent.KEYCODE_DPAD_LEFT -> ballX -= rnd.nextInt(50)
      KeyEvent.KEYCODE_DPAD_RIGHT -> ballX += rnd.nextInt(50)
      KeyEvent.KEYCODE_DPAD_UP -> ballY -= rnd.nextInt(50)
      KeyEvent.KEYCODE_DPAD_DOWN -> ballY += rnd.nextInt(50)
      KeyEvent.KEYCODE_SPACE -> {
         ballX += rnd.nextInt(100) - 50
         ballY += rnd.nextInt(100) - 50
      }
      else -> return false
    }
    invalidate()
    return true // event handled
}
```

Project:List.zip

Option Menu

(onCreateOptionMenu)

```
<menu
   xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:id="@+id/pause" android:icon="@drawable/pause“
        android:title="Pause">
   </item>
   <item android:id="@+id/play"  android:icon="@drawable/play"
        android:title="Play">
   </item>
   <item android:id="@+id/stop" android:icon="@drawable/stop"
        android:title="Stop">
   </item>
</menu>
```

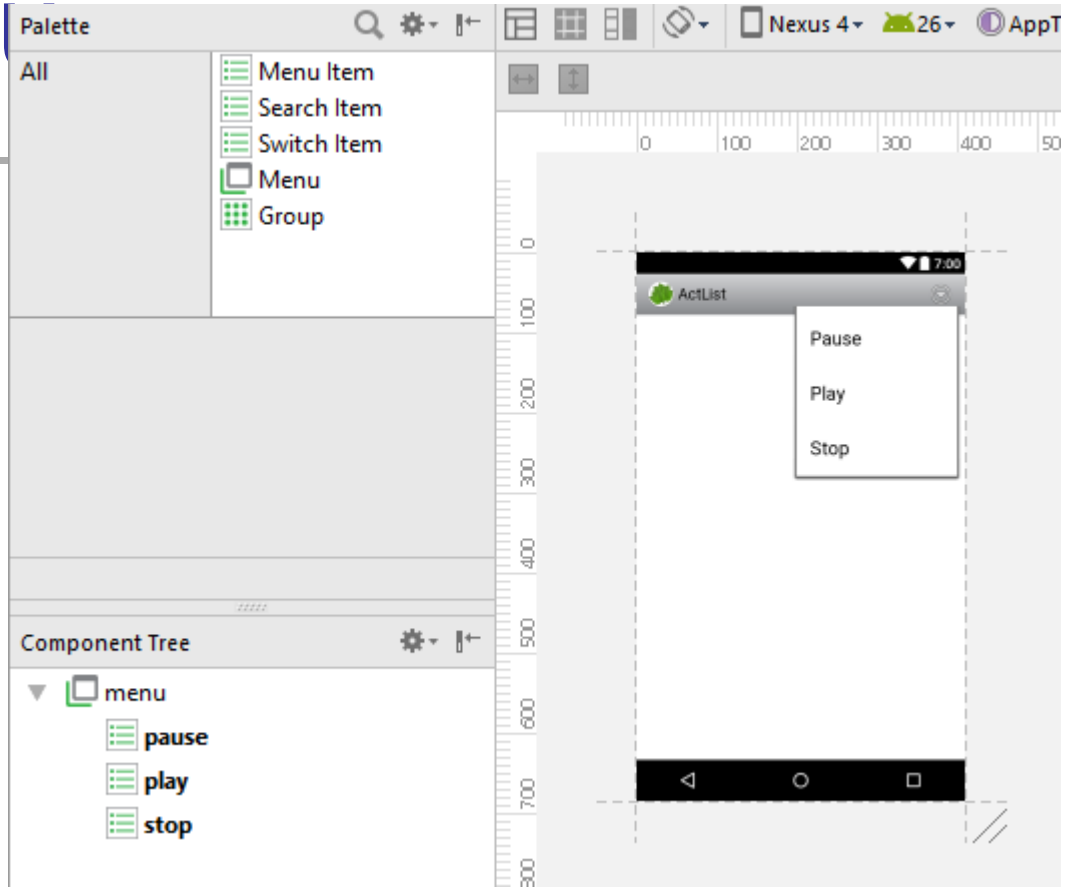
```
override fun onCreateOptionsMenu(menu: Menu): Boolean {
   val inflater = menuInflater
   inflater.inflate(R.menu.activity_canvas, menu)
   return super.onCreateOptionsMenu(menu)
}
```

Project:List.zip

Option Menu
(onCreateOptionMenu)

Rovnako dobre to môžete navrhovať v editore

Spôsob zobrazenia a renderovania závisí na API level zariadenia




```
<menu
     xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:id="@+id/pause" android:icon="@drawable/pause“                android:title="Pause">
     </item>
   <item android:id="@+id/play"  android:icon="@drawable/play"    android:title="Play">
     </item>
   <item android:id="@+id/stop" android:icon="@drawable/stop"     android:title="Stop">
     </item>
</menu>
```

```
@Override
```

Project:List.zip

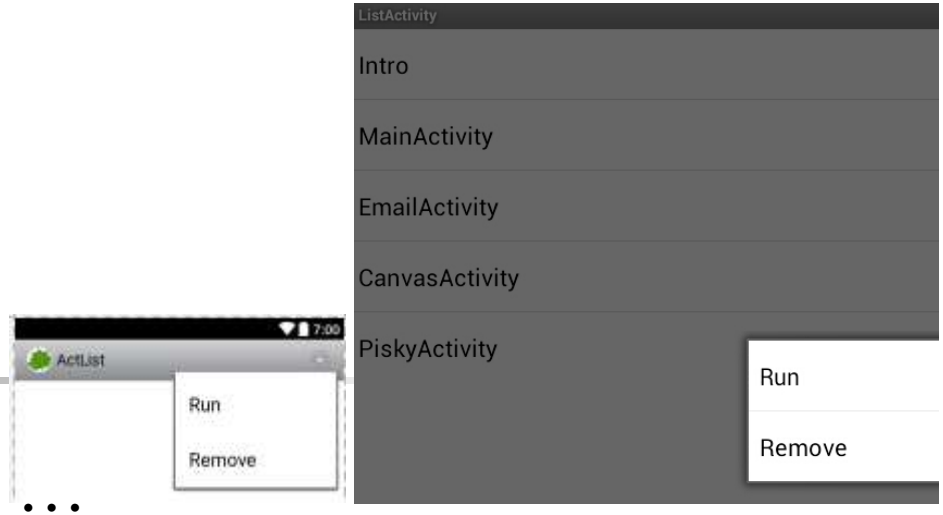
Option Menu

```
Thread th = new Thread() {
fun run() {
    while (!stopped) {
        if (!paused) {
            ...
```

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    when (item.getItemId()) {
        R.id.pause -> {
            canvasView1.paused = true
            return true
        }
        R.id.play -> {
            canvasView1.paused = false
            return true
        }
        R.id.stop -> {
            canvasView1.stopped = true
            return true
        }
        else -> return super.onOptionsItemSelected(item)
    }
}
```

Project:List.zip

Context Menu




```
Override fun onCreate(
   savedInstanceState:Bundle?) {  ...
   registerForContextMenu(listView1); // na rozdiel od
} // OptionMenu, ContextMenu treba registrovať k príslušnému view
override fun onCreateContextMenu(menu: ContextMenu?, v: View?,
                menuInfo: ContextMenu.ContextMenuInfo? ) {
     getMenuInflater().inflate(R.menu.list_menu, menu)
}
```

```
override fun onContextItemSelected(item: MenuItem): Boolean {
  val info = item.getMenuInfo() as AdapterContextMenuInfo
  val className = actList.get(info.id.toInt())
  when (item.getItemId()) {
     R.id.remove -> {
         actList.removeAt(info.id.toInt())
          la.notifyDataSetChanged()
         return true
     }
```

Project:List.zip

Maľovátko

(MotionEvent actions)

finger paint

```
private val mPath: Path
override protected fun onDraw(canvas: Canvas) {
   super.onDraw(canvas)
   canvas.drawPath(mPath, mPaint)
}
override fun onTouchEvent(event: MotionEvent): Boolean {
   val x = event.x
   val y = event.y
   when (event.action) {
     MotionEvent.ACTION_DOWN -> {
       startTouch(x, y) invalidate()   }
     MotionEvent.ACTION_MOVE -> {
       moveTouch(x, y)  invalidate()   }
     MotionEvent.ACTION_UP -> {
       upTouch() invalidate()          }
   }
   return true
}
```

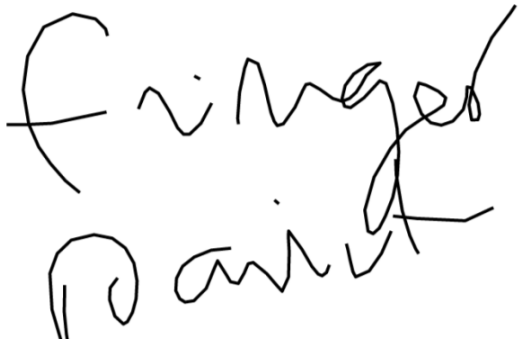
https://examples.javacodegeeks.com/android/core/graphics/canvas-graphics/android-canvas-example/

Project:CanvasExample.zip

AndroidCanvasExample

Maľovátko

(bezier vs. linear - nebezier)

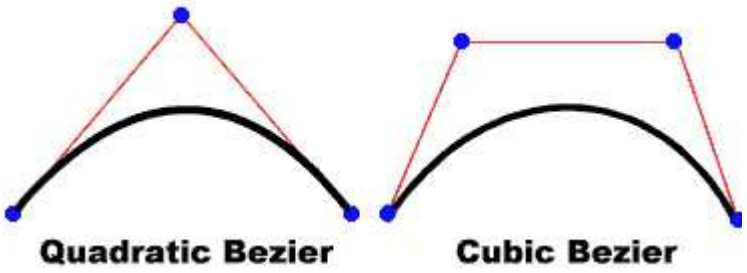



```
private fun startTouch(x: Float, y: Float) {
  mPath.moveTo(x, y)
  lastX = x
  lastY = y
}
private val TOLERANCE = 5f
private fun moveTouch(x: Float, y: Float) {
  val dx = Math.abs(x - lastX)
  val dy = Math.abs(y - lastY)
  if (dx >= TOLERANCE || dy >= TOLERANCE) {
     // mPath.quadTo(lastX, lastY, (x+lastX)/2, (y+lastY)/2)
     mPath.lineTo(x, y);
     lastX = x
     lastY = y
  }
}
```

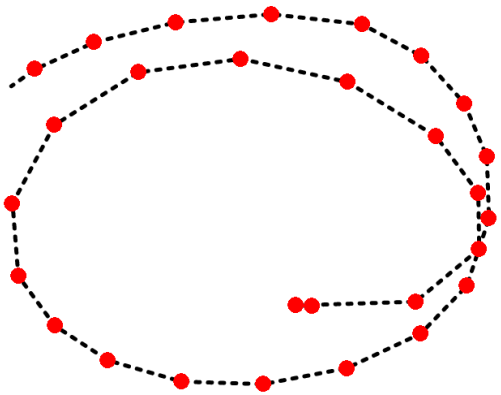
https://examples.javacodegeeks.com/android/core/graphics/canvas-graphics/android-canvas-example/

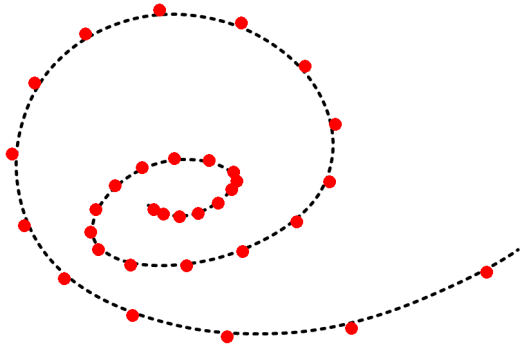
Project:CanvasExample.zip

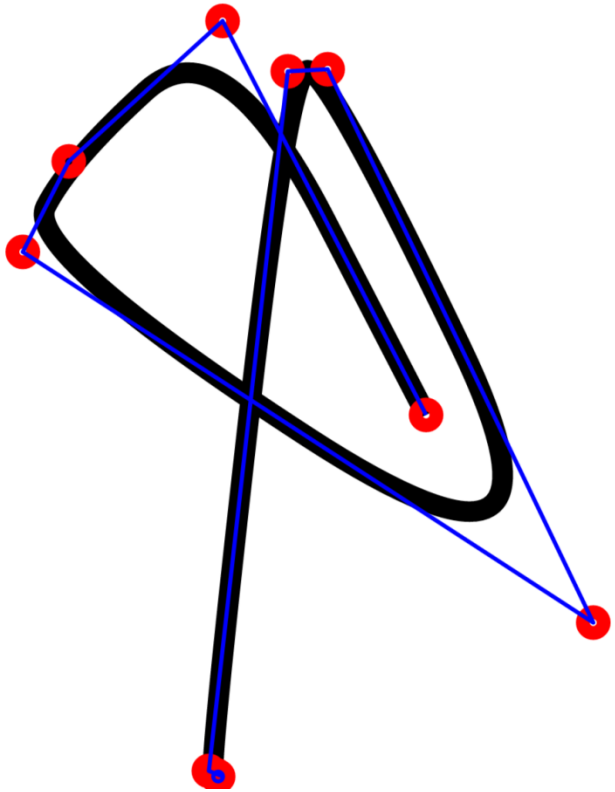
Bézier



- `lineTo(x,y)`
- `quadTo(controlX, controlY, x, y)`
- `cubeTo(controlX1, controlY1, controlX2, controlY2, x, y)`







Prémia

(na tému maľovátka)