File - C:\Users\maadmin\AndroidStudioProjects\a_spring_26_1\app\src\main\java\com\example\a_spring_26_1\MainActivity.kt

```kotlin
package com.example.a_spring_26_1

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.view.View

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v,
insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }
        val up1View = Up1View(this, null)
        setContentView(up1View)
        up1View.startAnimation()
    }
}

```

```kotlin
package com.example.a_spring_26_1

import android.content.Context
import android.graphics.Canvas
import android.graphics.Color
import android.graphics.Paint
import android.graphics.RectF
import android.os.Handler
import android.os.Looper
import android.util.AttributeSet
import android.view.MotionEvent
import android.view.View
import kotlin.math.atan
import kotlin.math.cos
import kotlin.math.sin
import kotlin.math.sqrt

class Up1View (context: Context, attrs: AttributeSet?) : View(context, attrs) {
    // var i: Int = 0;var j: Int =0
    var b: Float =0f;var h: Float =0f

    var x1: Float =0f
    var x2: Float =0f
    var x3: Float =0f
    var x4: Float =0f
    var x5: Float =0f
    var y1: Float =0f
```

```
28      var y2: Float =0f
29      var y3: Float =0f
30      var y4: Float =0f
31      var y5: Float =0f
32      var y6: Float =0f
33      var y7: Float =0f
34      var lp: Float =0f
35      //var rad: Float =0f
36      //var idx: Float =0f
37      //var idxx: Float =0f
38      //var idy: Float =0f
39      //var idyy: Float =0f
40      var ix0: Float =0f
41      var iy0: Float =0f
42      var ix1: Float =0f
43      var iy1: Float =0f
44      var ix2: Float =0f
45      var iy2: Float =0f
46
47      var str0: String =""
48
49      var irun: Int = 0; var iwisch: Int=0; var iall: Int =0
50      val nw: Int = 7201
51      var xun = DoubleArray(7201)
52      var xob = DoubleArray(7201)
53      var fbod = DoubleArray(7201)
54      var xkurb = DoubleArray(7201)
```

```
55      var xxkurb = DoubleArray(7201)
56      var m1: Double = 16.0
57      var m2: Double = 32.0
58      var c1: Double = 500000.0
59      var c2: Double = 50000.0
60      var ee: Double = 0.03
61      var g: Double = 9.81
62      var b1 = 2.0*0.4*sqrt ( c1*m1 )
63      var b2 = 2.0*0.15*sqrt ( c2*m2 )
64      var fre: Double = 11.0
65      var om=0.0;var dt=0.0;var fak=0.0;var fpleu=0.0
66      var xp_un=0.0;var xp_ob=0.0
67      //var scale0=3.0
68      var scale=0.0
69      var iyun: Float = 0f; var iyob: Float = 0f
70      var ispeed:Int = 2
71      var del_i: Int = 10; var ii: Int =7200-720-del_i
72      var speed: Long = 16L
73      var idrawline: Int = 0
74      var xdown = 0f
75      var ydown = 0f
76      var xup = 0f
77      var yup = 0f
78
79      var mypi= 4*atan(1.0)
80      //var winkel: Double=0.0
81      //var ixx1: Float = 0f
```

```
 82      //var iyy1: Float = 0f
 83      //var ixx2: Float = 0f
 84      //var iyy2: Float = 0f
 85
 86      private val paint = Paint().apply {
 87          color = Color.RED
 88          isAntiAlias = true
 89      }
 90
 91      private var ballRadius = 50f
 92      private var ballX = 200f
 93      private var ballY = 200f
 94      private var ySpeed = 20f // Geschwindigkeit in Y-Richtung
 95      private var xSpeed = 10f // Geschwindigkeit in X-Richtung (optional)
 96      private val handler = Handler(Looper.getMainLooper())
 97      private val runnable = object : Runnable {
 98          override fun run() {
 99              updateBallPosition()
100              invalidate() // View neu zeichnen
101              handler.postDelayed(this, speed) // vorher 16 //~60 FPS
102          }
103      }
104
105      override fun onDraw(canvas: Canvas) {
106          super.onDraw(canvas)
107          //var h=canvas.height
108          //var b=canvas.width
```

```
109          b=canvas.width.toFloat()
110          h=canvas.height.toFloat()
111          // eingefügt 10.09.2016 Hintergrundfarbe
112          paint.setColor(Color.rgb(255,240,40)) // Bomag gelb
113          canvas.drawRect(0f,0f,b.toFloat(),h.toFloat(),paint)
114          paint.setColor(Color.BLACK)
115
116          if(iall==1){
117              paint.setColor(Color.RED)
118              canvas.drawCircle(ballX, ballY, ballRadius, paint)
119              paint.textSize = 60f
120              paint.setColor(Color.DKGRAY)
121              canvas.drawText("ispeed "+ispeed.toString(),100f,140f,paint)
122              canvas.drawText("b "+b.toString()+" h "+h.toString(),100f,220f,paint)
123              canvas.drawText("xdown "+xdown.toString(),100f,300f,paint)
124              canvas.drawText("ydown "+ydown.toString(),100f,380f,paint)
125              canvas.drawText("xup "+xup.toString(),100f,460f,paint)
126              canvas.drawText("yup "+yup.toString(),100f,540f,paint)
127              paint.strokeWidth=10f
128              paint.setColor(Color.LTGRAY)
129              if(idrawline==1){
130                  canvas.drawLine(xdown.toFloat(),ydown.toFloat(),xup.toFloat(),yup.
    toFloat(),paint)
131              }
132              paint.setColor(Color.BLACK)
133              /*ixx1=400f
134              iyy1=400f
```

```
135             ixx2=ixx1+200f*cos((winkel*mypi/180.0).toFloat())
136             iyy2=iyy1-200f*sin((winkel*mypi/180.0).toFloat())
137             canvas.drawLine(ixx1,iyy1,ixx2,iyy2,paint)
138             winkel=winkel+1.0*/
139         }
140
141         if((iwisch==0)&&(h>b)){
142             paint.setColor(Color.BLACK)
143             paint.textSize = (15*b/480).toFloat()
144             str0 = "Explanation / Erläuterung:"
145             canvas.drawText(str0,(10*b/480).toFloat(),(40*h/800).toFloat(),paint)
146             str0="Wiping vertical: Animation faster/slower"
147             canvas.run { drawText(str0,(20*b/480).toFloat(),(55*h/800).toFloat(),
    paint) }
148             str0 = "Wischen vertikal: Animation schneller/langsamer"
149             canvas.drawText(str0,(20*b/480).toFloat(),(70*h/800).toFloat(),paint)
150             //paint.textSize = (15*b/480).toFloat()
151             str0 = "Explanations are deleted after first wiping action"
152             canvas.drawText(str0,(30*b/480).toFloat(),(85*h/800).toFloat(),paint)
153             str0 = "Erläuterungen werden gelöscht nach erster Wischaktion"
154             canvas.drawText(str0,(30*b/480).toFloat(),(100*h/800).toFloat(),paint)
155         }
156
157         if(irun==0){
158             dt=1.0/(720.0*fre)
159             om=2.0*mypi*fre
160             xun[0]=0.0;xun[1]=0.0
```

```
161             xob[0]=0.0;xob[1]=0.0
162             for(i in 0..nw-1){
163                 xkurb[i]=ee*sin(om*i*dt)
164                 xxkurb[i]=ee*cos(om*i*dt)
165             }
166             for(i in 1..(nw-2)){
167                 fpleu=c2*(xun[i]-xob[i]-xkurb[i])+m2*g
168                 xp_un=(xun[i]-xun[i-1])/dt
169                 xp_ob=(xob[i]-xob[i-1])/dt
170                 fbod[i]=-c1*xun[i]-b1*xp_un+(m1+m2)*g
171                 if(fbod[i]<0)fbod[i]=0.0
172                 fak=fbod[i]-m1*g-fpleu-b2*(xp_un-xp_ob)
173                 xun[i+1]=2*xun[i]-xun[i-1]+fak*dt*dt/m1
174                 fak=fpleu-m2*g-b2*(xp_ob-xp_un)
175                 xob[i+1]=2*xob[i]-xob[i-1]+fak*dt*dt/m2
176             }
177             irun++
178         }
179
180         ii=ii+del_i
181         if(ii>7200)ii=ii-720
182         iy0=(0.85*h).toFloat()
183         y1=(0.06*h).toFloat()
184         y2=(0.02*h).toFloat()
185         y3=(0.43*h).toFloat()
186         y4=(0.11*h).toFloat()
187         y5=(0.12*h).toFloat()
```

```
188         y6=(0.08*h).toFloat()
189         y7=(0.02*h).toFloat()
190         lp=(0.20*h).toFloat()
191         ix0=(0.50*b).toFloat()
192         x1=(0.25*b).toFloat()
193         x2=(0.30*b).toFloat()
194         x3=(0.08*b).toFloat()
195         x4=(0.18*b).toFloat()
196         x5=(0.05*b).toFloat()
197         scale=2.75*h
198
199         // Fuß
200         paint.setColor(Color.BLACK)
201         paint.strokeWidth=(5f*b/480f).toFloat()
202         if(h<b)paint.strokeWidth=(5f*h/480f).toFloat()
203         ix1=ix0-x1;iy1=iy0-y7-(xun[ii]*scale).toFloat();ix2=ix1+x5;iy2=iy1+y7
204         canvas.drawLine(ix1,iy1,ix2,iy2,paint)
205         ix1=ix2;iy1=iy2;ix2=ix0+x1-x5;iy2=iy1;canvas.drawLine(ix1,iy1,ix2,iy2,paint
    )
206         ix1=ix2;iy1=iy2;ix2=ix1+x5;iy2=iy1-y7;canvas.drawLine(ix1,iy1,ix2,iy2,paint
    )
207         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1+y7-y1;canvas.drawLine(ix1,iy1,ix2,iy2,paint
    )
208         ix1=ix2;iy1=iy2;ix2=ix1-2*x1;iy2=iy1;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
    // horizontal
209         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1+y1-y7;canvas.drawLine(ix1,iy1,ix2,iy2,paint
    )
```

```
210         // Linie vom Fu  nach oben gehend
211         ix1=ix0;iy1=iy0-y1-(xun[ii]*scale).toFloat()
212         ix2=ix0;iy2=iy0-y1-y2-(xun[ii]*scale).toFloat();canvas.drawLine(ix1,iy1,ix2
    ,iy2,paint)
213         iyun=iy2
214         // Oberbau
215         ix1=ix0-x2;iy1=iy0-y1-y3-(xob[ii]*scale).toFloat();ix2=ix0-x4;iy2=iy1
216         canvas.drawLine(ix1,iy1,ix2,iy2,paint)
217         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1-y6;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
218         ix1=ix2;iy1=iy2;ix2=ix0+x4;iy2=iy1;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
219         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1+y6;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
220         ix1=ix2;iy1=iy2;ix2=ix0+x2;iy2=iy1;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
221         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1-y5;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
222         ix1=ix2;iy1=iy2;ix2=ix1-2*x2;iy2=iy1;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
223         ix1=ix2;iy1=iy2;ix2=ix1;iy2=iy1+y5;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
224         // Linie vom Oberbau nach unten gehend
225         ix1=ix0;iy1=iy0-y1-y3-y6-(xob[ii]*scale).toFloat()
226         ix2=ix1;iy2=iy1+y4;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
227         // Kurbel
228         ix1=ix2;iy1=iy2;ix2=ix1+(xxkurb[ii]*scale).toFloat();iy2=iy1-(xkurb[ii]*
    scale).toFloat()
229         canvas.drawLine(ix1,iy1,ix2,iy2,paint)
230         // Pleuel (langes Pleuel ee<<lp)
231         ix1=ix2;iy1=iy2;ix2=ix0;iy2=iy1+lp;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
232         iyob=iy2
233         // Feder, von oben nach unten
234         ix1=ix0;iy1=iyob;ix2=ix0+x3;iy2=(0.875*iyob+0.125*iyun).toFloat();canvas.
```

```
234 drawLine(ix1,iy1,ix2,iy2,paint)
235         ix1=ix2;iy1=iy2;ix2=ix0-x3;iy2=(0.625*iyob+0.375*iyun).toFloat();canvas.
    drawLine(ix1,iy1,ix2,iy2,paint)
236         ix1=ix2;iy1=iy2;ix2=ix0+x3;iy2=(0.375*iyob+0.625*iyun).toFloat();canvas.
    drawLine(ix1,iy1,ix2,iy2,paint)
237         ix1=ix2;iy1=iy2;ix2=ix0-x3;iy2=(0.125*iyob+0.875*iyun).toFloat();canvas.
    drawLine(ix1,iy1,ix2,iy2,paint)
238         ix1=ix2;iy1=iy2;ix2=ix0;iy2=iyun;canvas.drawLine(ix1,iy1,ix2,iy2,paint)
239         // Bodenkraft
240         paint.setColor(Color.GREEN)
241         fak=fbod[ii]/((m1+m2)*g)*(0.01*h).toFloat()
242         ix1=ix0;iy1=iy0-(xun[ii]*scale).toFloat();ix2=ix1;iy2=iy1+fak.toFloat()
243         //canvas.drawLine(ix1-150,iy1,ix2-150,iy2,paint)
244         val ovalr =
245             RectF(ix1 - x1 + x5, iy1 + 3 - (fak).toInt(), ix1 + x1 - x5, iy1 + 3
     + (fak).toInt())
246         canvas.drawArc(ovalr,0f,180f, true, paint)
247     }
248
249     private fun updateBallPosition() {
250         ballY += ySpeed
251         ballX += xSpeed
252
253         // Kollisionserkennung: Boden
254         if (ballY + ballRadius > height) {
255             ballY = height - ballRadius
256             ySpeed = -ySpeed
```

```
257         }
258         // Kollisionserkennung: Decke
259         else if (ballY - ballRadius < 0) {
260             ballY = ballRadius
261             ySpeed = -ySpeed
262         }
263
264         // Kollisionserkennung: Seiten (optional)
265         if (ballX + ballRadius > width || ballX - ballRadius < 0) {
266             xSpeed = -xSpeed
267         }
268     }
269
270     fun startAnimation() {
271         handler.post(runnable)
272     }
273
274     fun stopAnimation() {
275         handler.removeCallbacks(runnable)
276     }
277
278     override fun onTouchEvent(event: MotionEvent): Boolean {
279         when (event.action){
280             MotionEvent.ACTION_DOWN -> {
281                 xdown = event!!.x
282                 ydown = event!!.y
283                 idrawline=0
```

```
284                 return true
285             }
286             MotionEvent.ACTION_UP -> {
287                 xup = event!!.x
288                 yup = event!!.y
289                 idrawline=1
290                 iwisch=1
291                 if((yup-ydown)>200){
292                     ispeed--
293                     if(ispeed<1)ispeed=1
294                 }
295                 if((yup-ydown)<-200){
296                     ispeed++
297                     if(ispeed>5)ispeed=5
298                 }
299                 if(ispeed==1){speed=32;del_i=10}
300                 if(ispeed==2){speed=16;del_i=10}
301                 if(ispeed==3){speed=8;del_i=10}
302                 if(ispeed==4){speed=8;del_i=20}
303                 if(ispeed==5){speed=8;del_i=40}
304
305                 if((xup-xdown)>0.75*b){
306                     if((-yup+ydown)>0.75*h){
307                         iall=1
308                     }
309                 }
310                 return true
```

```
311                 }
312             }
313
314             invalidate()
315             return super.onTouchEvent(event)
316
317         }
318 }
```