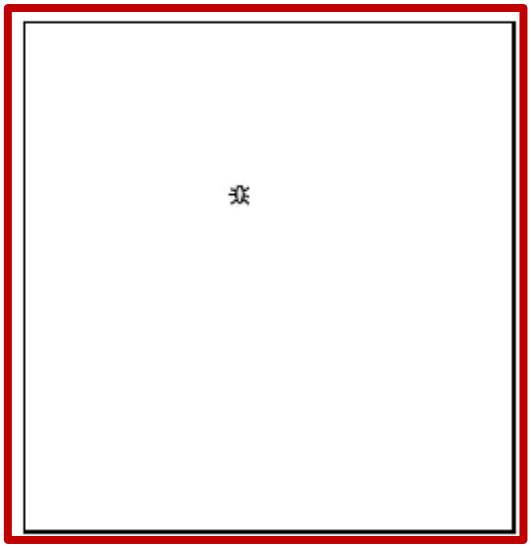
RADIO VADĀMA PAVASARA MUŠA
30.nodarbība

IEPRIEKŠĒJĀ NODARBĪBĀ

- Iepriekšējā nodarbībā – muša, kas kustas pati.
- Mums jau ir:
  - vizuālā komponente,
  - pozīcijas pārbaude,
  - strādnieks.

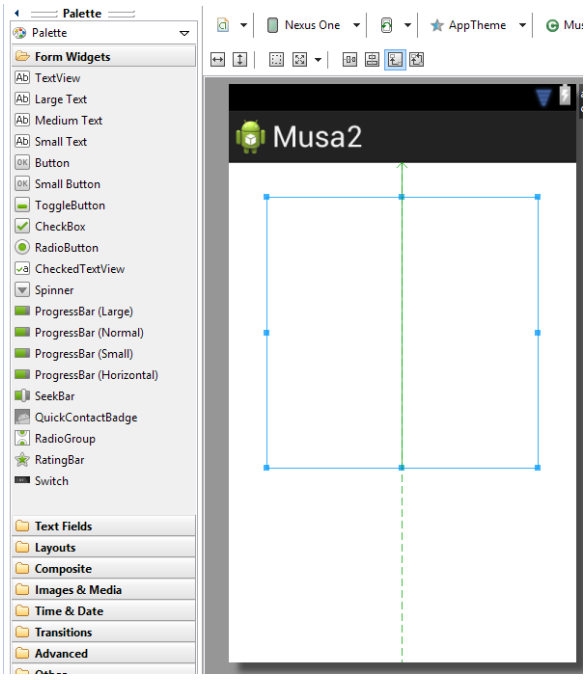


ŠAJĀ NODARBĪBĀ

- Veidosim mušu, kuru var pārvietot ar pogām.
- Darbu saraksts:
  - jāpievieno pogas,
  - jānoņem strādnieks,
  - jāizveido kustība.

ESOŠAIS EKRĀNS

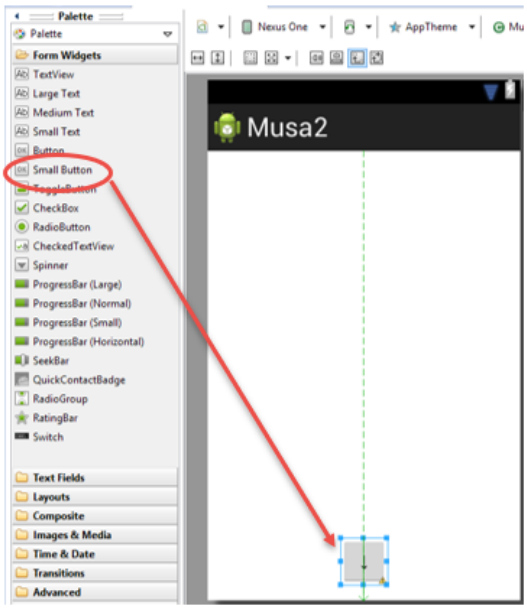
- Ekrāna augšā – vizuālā komponente.



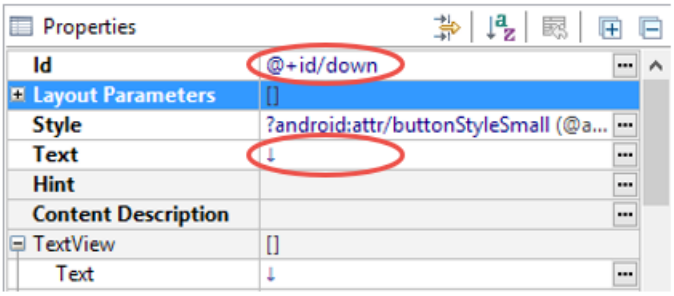
POGAS «↓» PIEVIENOŠANA

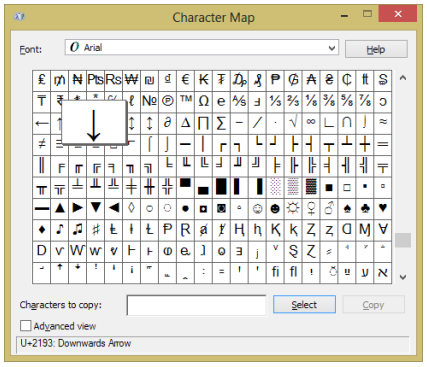
Poga «↓»:

- ekrāna apakšā, centrā



- Id = @+id/down
- Text = ↓
- On Click = down



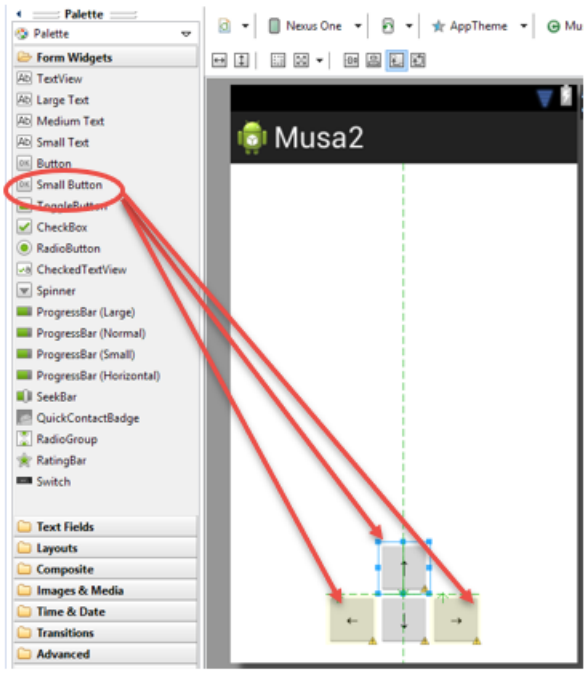


PĀRĒJO POGU PIEVIENOŠANA

- Poga «↑»:
  - Id = @+id/up
  - Text = ↑
  - On Click = up

- Poga «←»:
  - Id = @+id/left
  - Text = ←
  - On Click = left

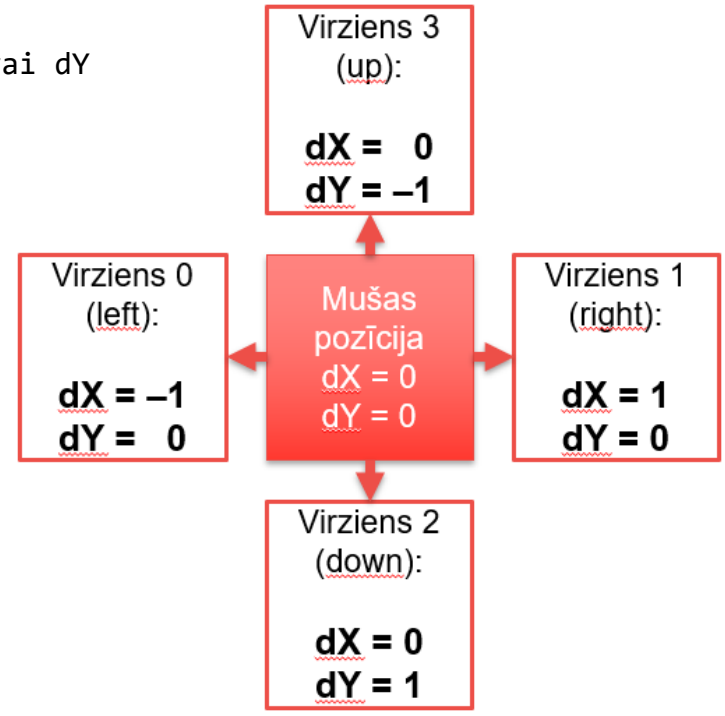
- Poga «→»:
  - Id = @+id/right
  - Text = →
  - On Click = right



Katrai pogai pievieno parametrus:
- Min Height = 60dp
- Min Width = 60dp

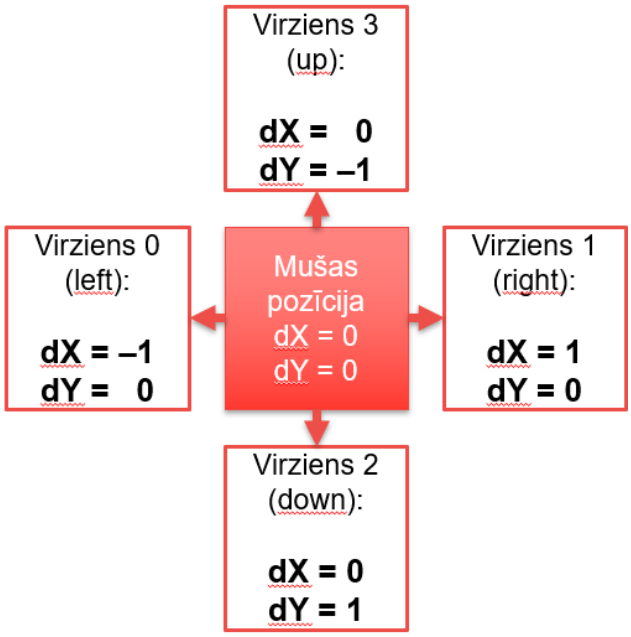
VIRZIENS

Atkarībā no virziena maina dX vai dY



METODE, KAS MAINA VIRZIENU

```
public class MusaView extends View{
. . .
  // virziens
  int dX = 0; // -1, ja pa kreisi un +1, ja pa labi
  int dY = -1; // -1, ja uz augšu un +1, ja uz leju
. . .
  public void changeDirection(int direction){
    // noņem virzienu
    dX = 0;
    dY = 0;
    switch (direction) {
      case 0: // pa kreisi
        dX = -1;break;
      case 1: // pa labi
        dX = 1;break;
      case 2: // uz leju
        dY = 1;break;
      case 3: // uz augšu
        dY = -1;break;
      default:
        break;
    }
  }
. . .
}
```



METODE MOVE()

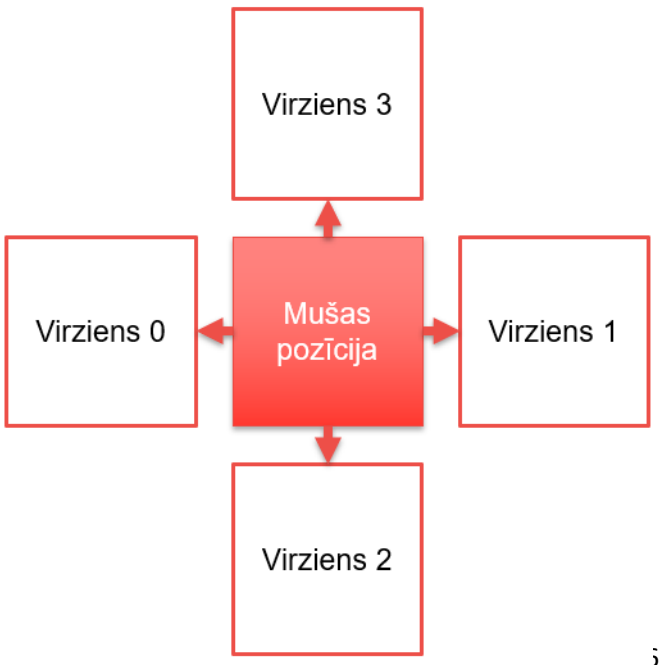
```
public class MusaView extends View{
. . .
  public void move() {

    int direction = r.nextInt(4);
    switch (direction) {
      case 0:
        x--; break;
      case 1:
        x++; break;
      case 2:
         y--; break;
      case 3:
        y++; break;
      default:
        break;
    }


    x = x + dX;
    y = y + dY;

  }
. . .
}
```

PAGAIDĀM APTUR STRĀDNIEKU

```
public class MusaActivity extends Activity {
   MusaView musa;
   //Handler stradnieks;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_musa);
     musa = (MusaView)findViewById(R.id.musaView);
     musa.prepareForDrawing();
     // apturam strādnieku
     // stradnieks = new Handler();
     // stradnieks.postDelayed(uzdevums, 1000);
   }
   public void update(){
     musa.move();
     if (musa.validPosition()){
       musa.invalidate(); // invalidate will invoke onDraw method in near future
       //stradnieks.postDelayed(uzdevums, 200); //gribam atjaunot ekrānu 5x sekundē
    }else{
       Toast.makeText(getBaseContext(), "Muša beigta", Toast.LENGTH_SHORT).show();
     }
  }
}
```

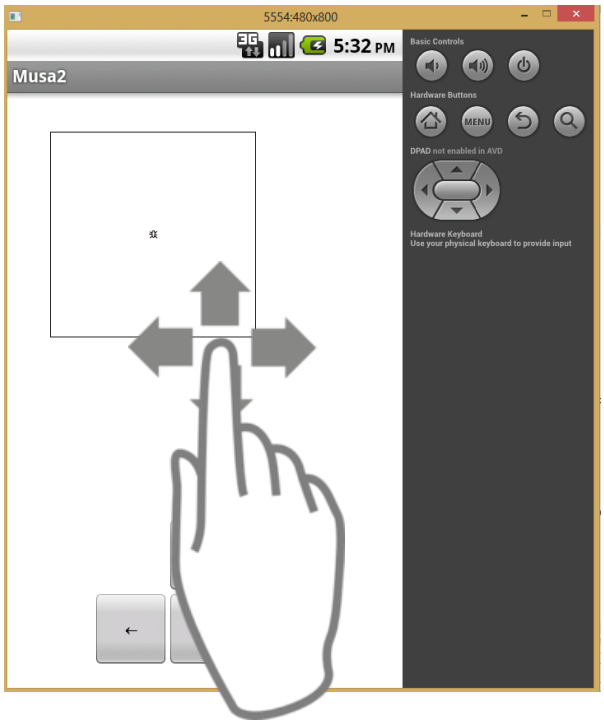
POGU METODES

```
public class MusaActivity extends Activity {
. . .
  public void up(View v){
    musa.changeDirection(3);
    update();
  }
  public void down(View v){
    musa.changeDirection(2);
    update();
  }
  public void left(View v){
    musa.changeDirection(0);
    update();
  }
  public void right(View v){
    musa.changeDirection(1);
    update();
  }
. . .
}
```

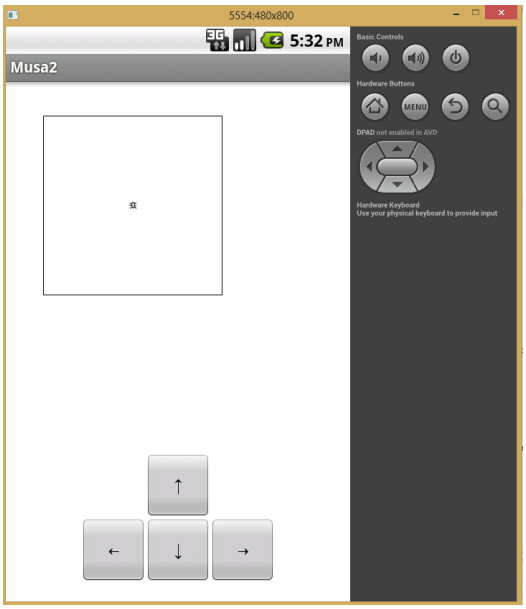


;

TESTĒJAM!



PAVILKŠANA (SWIPE, FLING)



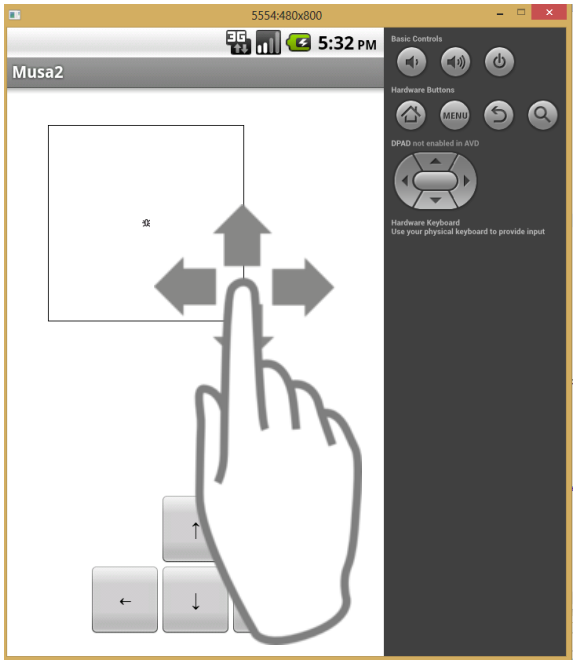
ŽESTA «FLING» PIEVIENOŠANA

```
public class Musa2 extends Activity {
. . .
  private GestureDetectorCompat mDetector;
. . .
  protected void onCreate(Bundle savedInstanceState) {
. . .
      mDetector = new GestureDetectorCompat(this,
                          new MyGestureListener());
. . .
  }
. . .
  @Override
  public boolean onTouchEvent(MotionEvent event){
       this.mDetector.onTouchEvent(event);
       return super.onTouchEvent(event);
    }
. . .
}
```

ŽESTA APSTRĀDE

```
public class Musa2 extends Activity {
. . .
  class MyGestureListener extends GestureDetector.SimpleOnGestureListener {
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
      int fx, fy;
      fx = (int) (e2.getX() - e1.getX()); // horizontālais attālums
      fy = (int) (e2.getY() - e1.getY()); // vertikālais attālums
      if( Math.abs(fx) > Math.abs(fy)) // noskaidro, kurā virzienā vairāk braukts ar pirkstu
      { // fling X
        if( fx < -30 ) { // left
          musa.changeDirection(0);
          update();
        }
        if( fx > 30 ) { // right
          musa.changeDirection(1);
          update();
        }
      } else { // fling Y
        if( fy < -30 ) { // up
          musa.changeDirection(3);
          update();
        }
        if( fy > 30 ) { // down
          musa.changeDirection(2);
          update();
        }
      }
      return true;
    }
  } // class MyGestureListener
} // Activity
```

TESTĒJAM PĀRVILKŠANU



IZMANTOTI ATTĒLI NO

- http://www.clker.com/clipart-swipe-all-directions.html