

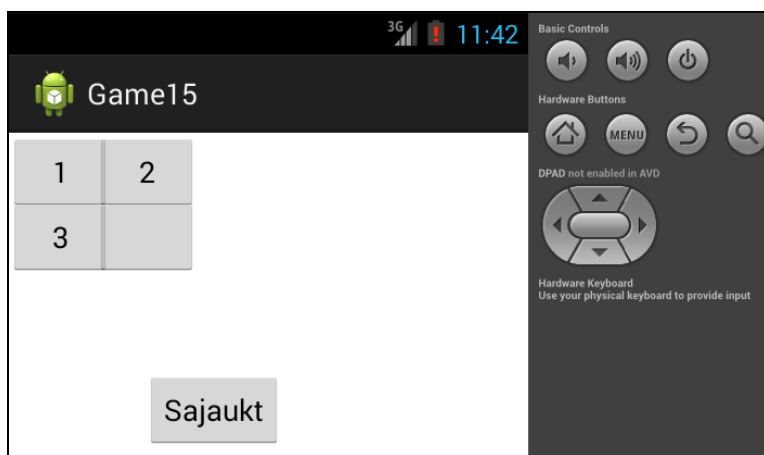
## SPĒLE 16 – PROGRAMMAS IZVEIDE

### 26.nodarbība

#### MĒRĶIS – GATAVA SPĒLES PROGRAMMA

Uzdevumi:

- Kā reakciju uz skaitļa pogas klikšķi pievienot skaitļa pārbīdi.
- Kā reakciju uz pogas «Sajaukt» klikšķi pievienot skaitļu sajaukšanu.
- Pievienot metodi, kas pārbauda, vai skaitļi ir pareizi sabīdīti.



#### KĀ PĀRBĪDĪT SKAITĻUS?

Vajadzēs atcerēties iepriekšējās nodarbības:

- kā izjaucām saliktu spēles laukumu,
- kā pogām piešķirām birku.

```
for(int rinda=0;rinda<N;rinda++){
    for(int kolonna=0;kolonna<N;kolonna++){
        Button poga=new Button(this);
        poga.setTag(rinda*N+kolonna);
        poga.setText((rinda*N+kolonna+1)+"");
        poga.setX(kolonna*80);poga.setWidth(40);
        poga.setY(rinda*60);poga.setHeight(30);
        poga.setOnClickListener(pogasReakcija);
        lauks[rinda][kolonna]=poga;
        this.addView(poga,izkartojums);
    }
}
```

```

int x0=3, y0=3, i=0;
while(i<10|game[3][3]!=0){
    int d=generator.nextInt(4);
    int dx=0, dy=0;
    switch (d) {
    case 0:
        dy=-1; break;
    case 1:
        dx=1; break;
    case 2:
        dy=1; break;
    case 3:
        dx=-1; break;
    }
    int xn=x0+dx, yn=y0+dy;
    if(xn>=0 && xn<4 && yn>=0 && yn<4){
        game[x0][y0]=game[xn][yn];
        game[xn][yn]=0;
        x0=xn; y0=yn;
        i++;
    }
}

```

## KĀDS MUMS LABUMS NO BIRKĀM?

|   | 0      | 1      | 2      | 3      |
|---|--------|--------|--------|--------|
| 0 | 1 (0)  | 2 (1)  | 3(2)   | 4(3)   |
| 1 | 5(4)   | 6(5)   | 7(6)   | 8(7)   |
| 2 | 9(8)   | 10(9)  | 11(10) | 12(11) |
| 3 | 13(12) | 14(13) | 15(14) | (15)   |

- Pieņemsim, ka spēles sākumā mēs izvēlamies pārbīdīt kauliņu ar skaitli 12.
- Tātad programmā tiks izdarīts klikšķis uz pogas, kurai ir pielikta birka 11.
- Mazliet aritmētikas, un no birkas vērtības varam iegūt tās rindas un kolonnas numuru, kurā atrodas poga:
  - uzklikšķinātā rinda  $x=11/4=2$
  - uzklikšķinātā kolonna  $y=11\%4=3$
- Atliek samainīt vietām tukšo un uzklikšķināto pogu.

## POGU SAMAINĪŠANA

- Mēs pogu nepārvietosim, bet nomainīsim tekstu uz pogas.
- Pogas skaitli nodzēsīsim un uzrakstīsim uz tukšās pogas.
- Atcerēsīsimies, kur tagad stāv tukšā poga.
- Svarīgi to izdarīt pareizā secībā.

```
lauks[tuksaRinda][tuksaKolonna].setText(lauks[uzklikskinataRinda][uzklikskinataKolonna].getText());  
lauks[uzklikskinataRinda][uzklikskinataKolonna].setText("");  
tuksaRinda=uzklikskinataRinda; tuksaKolonna=uzklikskinataKolonna;
```

## JA SPĒLĒTĀJS NEZINA NOTEIKUMUS?

- Programmai jāpārlicinās, ka izvēlētais gājiens ar kauliņu x;y ir atļauts.
- Neatļauts ir gājiens ar kauliņu, kuram blakus neatrodas tukšais kauliņš.
- Jānoskaidro, vai x;y un tukšais lauciņš x0;y0 ir kaimiņi.
- Kaimiņi ir tie, kas atrodas blakus - vienā rindā vai kolonnā.
- Ja vienā rindā, tad x0=x, bet y0=y-1 vai arī y0=y+1
- Ja vienā kolonnā, tad y0=y, bet x0=x-1 vai arī x0=x+1

|   |   |   |
|---|---|---|
| 2 |   | 3 |
| 1 | 4 | 5 |
| 7 | 8 | 6 |

NEATĻAUT!

## METODE, KAS PĀRBAUDA, VAI GĀJIENS IR ATĻAUTS

- Atveram iesākto projektu [Game15](#)
- Papildināsīm datni [Activity15.java](#) ar jaunu metodi:
  - `boolean` tipa funkciju `gajiensAtlauts()`
  - funkcijai dosim līdz 2 lielumus - uzklikšķinātā skaitļa pogas rindas un kolonnas numuru.

```
public boolean gajiensAtlauts(int rinda,int kolonna){
    if((tuksaRinda==rinda&&(tuksaKolonna==kolonna-1||tuksaKolonna==kolonna+1))||
        (tuksaKolonna==kolonna&&(tuksaRinda==rinda-1||tuksaRinda==rinda+1)))return true;
    else return false;
}
```

## POGAS REAKCIJAS PAPILDINĀŠANA

- Projektā jau ir *View.OnClickListener* tipa metode *pogasReakcija*, kas prot nolasīt uzklikšķinātās pogas birku jeb numuru:

```
View.OnClickListener pogasReakcija=new View.OnClickListener() {
    @Override
    public void onClick(View uzklikskinataisElements) {
        Button poga=(Button)uzklikskinataisElements;
        int pogasNr=Integer.parseInt(poga.getTag().toString());
        Toast.makeText(getApplicationContext(), pogasNr+"", Toast.LENGTH_SHORT).show();
    }
};
```

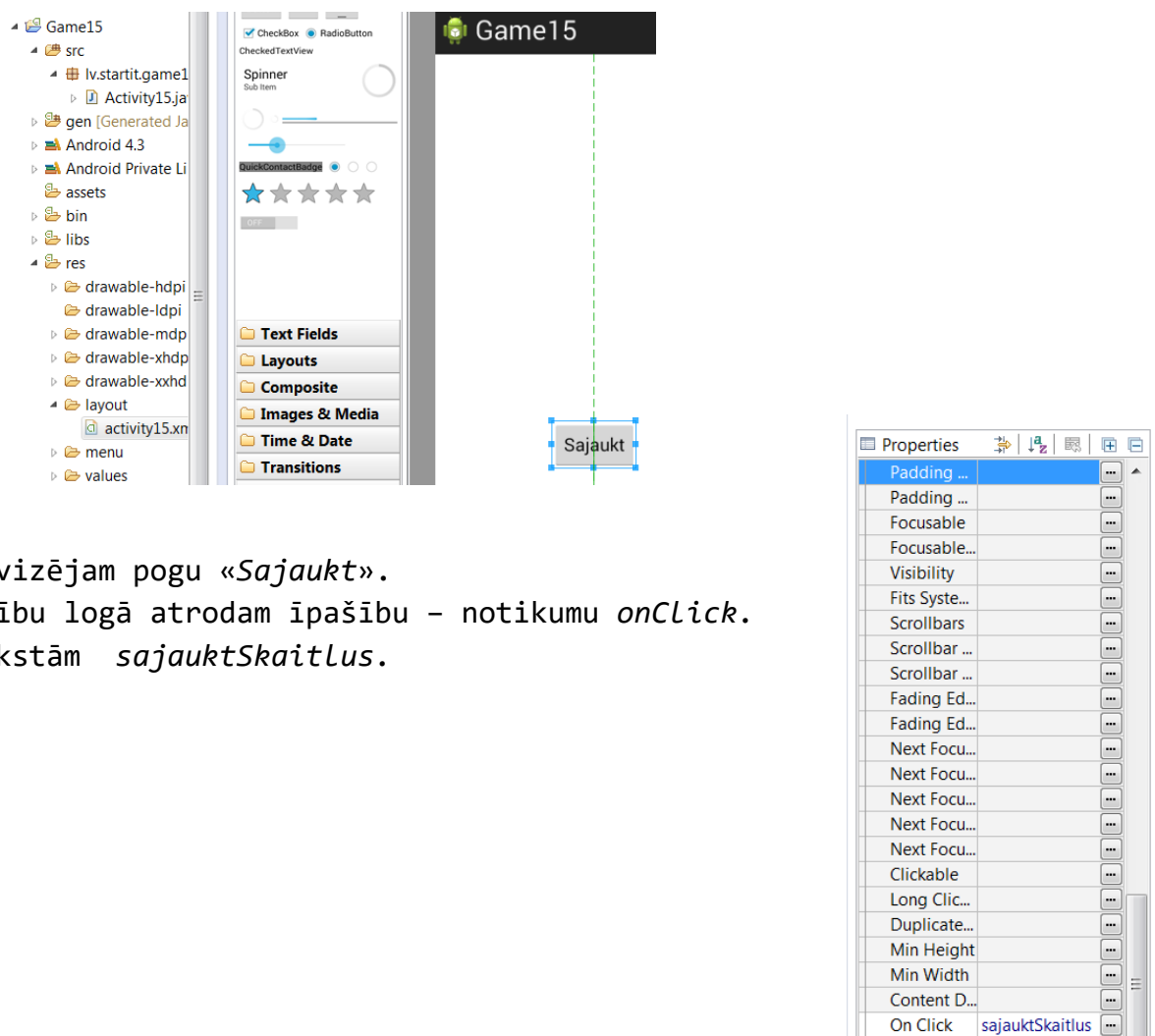
- Programmas tekstā aizvietosim rindiņu, kas parāda *Toast* uzrakstu, ar kodu, kas:
  - nosaka pogas rindu un kolonnu;
  - pārbauda, vai gājiens ir atļauts, t.i., vai tukšā poga atrodas blakus rindā vai kolonnā,
  - ja gājiens atļauts, tad veic šo gājienu.

```
View.OnClickListener pogasReakcija=new View.OnClickListener() {
    @Override
    public void onClick(View uzklikskinataisElements) {
        Button poga=(Button)uzklikskinataisElements;
        int pogasNr=Integer.parseInt(poga.getTag().toString());
        Toast.makeText(getApplicationContext(), pogasNr+"", Toast.LENGTH_SHORT).show();
    }
};
```

```
View.OnClickListener pogasReakcija=new View.OnClickListener() {
    @Override
    public void onClick(View uzklikskinataisElements) {
        Button poga=(Button)uzklikskinataisElements;
        int pogasNr=Integer.parseInt(poga.getTag().toString());
        int uzklikskinataRinda=pogasNr/N;
        int uzklikskinataKolonna=pogasNr%N;
        if(gajiensAtlauts(uzklikskinataRinda,uzklikskinataKolonna)){
            lauks[tuksaRinda][tuksaKolonna].setText(lauks[uzklikskinataRinda][uzklikskinataKolonna].getText());
            lauks[uzklikskinataRinda][uzklikskinataKolonna].setText("");
            tuksaRinda=uzklikskinataRinda; tuksaKolonna=uzklikskinataKolonna;
        }
    }
};
```

## POGAS «SAJUKT» APMĀCĪŠANA

- Projekta *Game15* mapēs atrodam dizaina veidni *activity15.xml* un atveram to.



- Aktivizējam pogu «*Sajaukt*».
- Īpašību logā atrodam īpašību – notikumu *onClick*.
- Ierakstām *sajauktSkaitlus*.

## METODE «sajauktSkaitlus»

- Papildināsim datni *Activity15.java* ar jaunu metodi *sajauktSkaitlus()*

```
public void sajauktSkaitlus(View v){
    for(int rinda=0;rinda<N;rinda++){
        for(int kolonna=0;kolonna<N;kolonna++){
            lauks[rinda][kolonna].setText((rinda*N+kolonna+1)+"");
        }
    }
    Random generator=new Random();
    int x0=N-1,y0=N-1,i=0;
    while(i<100||!(lauks[N-1][N-1].getText().toString().equals(""))){
        int d=generator.nextInt(4);
        int dx=0,dy=0;
        switch(d){
            case 0:dy=-1;break;
            case 1:dx=1;break;
            case 2:dy=1;break;
            case 3:dx=-1;break;
        }
        int xn=x0+dx; int yn=y0+dy;
        if(xn>=0&&xn<N&&yn>=0&&yn<N){
            lauks[x0][y0].setText(lauks[xn][yn].getText());
            lauks[xn][yn].setText("");
            x0=xn;y0=yn;
            i++;
        }
    }
    tuksaKolonna=N-1; tuksaRinda=N-1;
    if (irSalikts())sajauktSkaitlus(v);
}
```

## SPĒLES BEIGU STĀVOKLIS

- Kā noskaidrot, vai spēle ir galā - visi skaitļi ir pareizi salikti?
- Jāpārbauda:
  - vai pogas skaitlis atbilst birkai - ir par 1 lielāks?
  - vai tukšais lauciņš atrodas labajā apakšējā stūrī?

## PĀRBAUDE - VAI SKAITĻI SALIKTI?

- Papildināsim datni *Activity15.java* ar jaunu metodi - boolean tipa funkciju *irSalikts()*

```
public boolean irSalikts(){
    if (!lauks[N-1][N-1].getText().toString().equals(""))return false;
    for(int rinda=0;rinda<N;rinda++){
        for(int kolonna=0;kolonna<N;kolonna++){
            if(rinda!=N-1&&kolonna!=N-1&&
                !lauks[rinda][kolonna].getText().toString().equals((rinda*N+kolonna+1)+""))return false;
        }
    }
    return true;
}
```

- Ja tukšais lauciņš nav vietā, tad skaitļus nemaz nav vērts pārbaudīt.

```
public boolean irSalikts(){
    if (!lauks[N-1][N-1].getText().toString().equals(""))return false;
    for(int rinda=0;rinda<N;rinda++){
        for(int kolonna=0;kolonna<N;kolonna++){
            if(rinda!=N-1&&kolonna!=N-1&&
                !lauks[rinda][kolonna].getText().toString().equals((rinda*N+kolonna+1)+""))return false;
        }
    }
    return true;
}
```

- Ja tukšais lauciņš vietā, ejam cauri visām pogu masīva rindām un kolonnām un pārbaudām, vai pogas skaitlim atbilst pogas birka.

```
public boolean irSalikts(){
    if (!lauks[N-1][N-1].getText().toString().equals(""))return false;
    for(int rinda=0;rinda<N;rinda++){
        for(int kolonna=0;kolonna<N;kolonna++){
            if(rinda!=N-1&&kolonna!=N-1&&
                !lauks[rinda][kolonna].getText().toString().equals((rinda*N+kolonna+1)+""))return false;
        }
    }
    return true;
}
```

**SPĒLE IR GATAVA!**