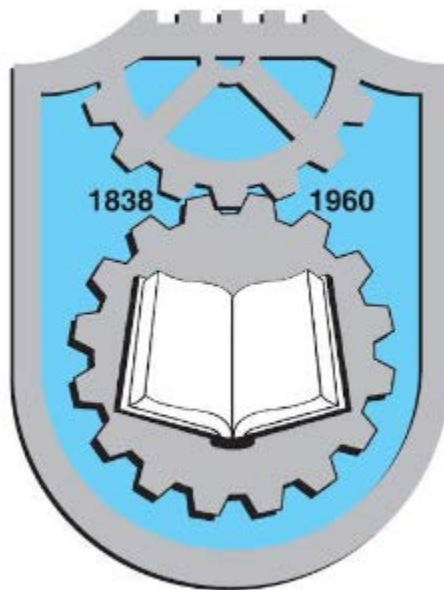


Универзитет у Крагујевцу
Факултет инжењерских наука



Пројекат:

Апликација за решавање Судоку пузле

Студент:

Урош Милошевић 628/2019

Предметни професор:

Др Вукашин Славковић

Садржај

Увод.....	3
Опис пројекта.....	3
Софтверска реализација система	6
Tabla.java	6
Activity_main.xml.....	10
LogikaResenja.java	11
Хардверска реализација система	14
Литература	15

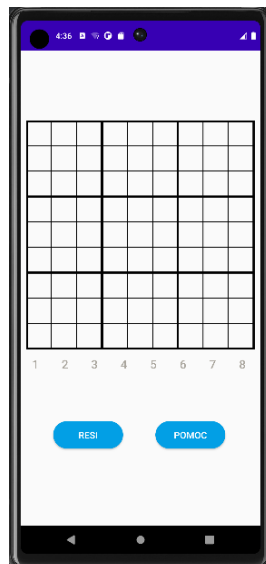
Увод

Сведоци смо велике експанзије паметних уређаја. Готово да нема појединца који у свом поседу нема бар један паметни уређај, а најчешће је у питању мобилни телефон. Андроид оперативни систем ја најзаступљенији међу свим оперативним системима. Апликације на Андроид уређијма су разноврсне, од оних које служе као системски алати до оних које нам олакшавају свакодневницу. Међутим, забава је огроман део индустрије, тако да су апликације тог типа јако битан фактор на целом тржишту. Пројекат којим се ми бавимо је управо тог типа, сврха му је омогућити једноставан алат који ће помоћи играчима у решавању Судоку пузле.[1]

Опис пројекта

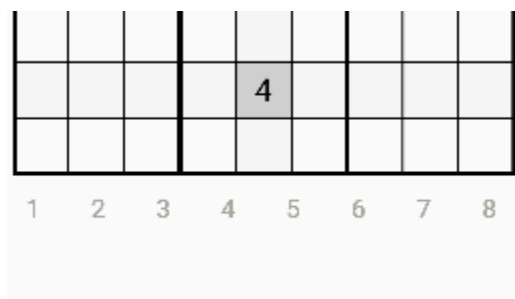
Пројектни задатак представља апликацију креирану у програмском језику Јава уз Помоћ Андроид Студио развојног окружења. Апликација функционише по једноставном принципу, без непотребних менија, при самом отварању апликације нам је доступан комплетан садржај апликације.

При покретању апликације кориснику се приказује празна Судоку табла.(слика 2.1.1)



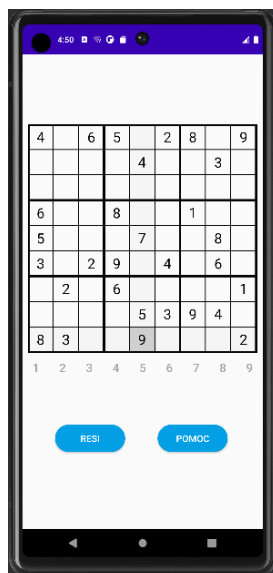
Слика 2.1.1 : Одабир корисника

Унос бројева у Судоку таблу се врши на следећи начин. Кликом на ћелију матрице која представља таблу се одабрано поље истиче, а кликом на број из понуђене селекције која се налази испод Судоку табле се уноси конкретна цифра у таблу.



Слика 2.2 : Унос цифара у таблу

Кликом на исту цифру која се налази у обележеној ћелији бришемо цифру из поља.

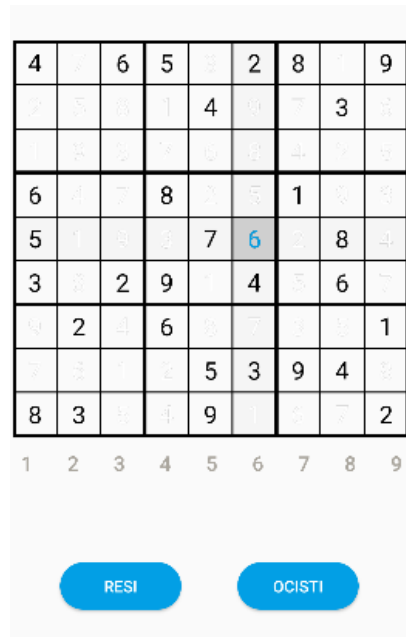


Слика 2.3: Пример попуњене табеле спремне за решавање

Опције које се нуде кориснику су:

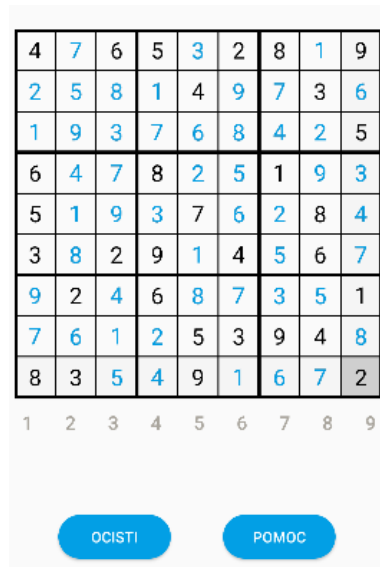
- Реши
- Помоћ

Кликом на дугме „Помоћ“ кориснику апликација помаже при решавању Судоку пузле тако што ће му открити поље које корисник обележи.



Слика 2.4 : Демонстрација
функције помоћи

Кликом на дугме „Реши“ ће нам се открити решење целе пазле, наглашене плавом бојом.



Слика 2.5 : Демонстрација
функције решавања

Кликом на дугме „Очисти“ се целокупан садржај табле брише и цела операција може бити отпочета изнова.

Софтверска реализација система

Апликација је реализована помоћу развојног окружења Андроид студио. У даљем тексту ћемо проћи кроз кључне подухвате у реализовању апликације

Tabla.java

Најпре ћемо прегазити onMeasure методу да бисмо добили ширину конкретног уређаја на којем се апликација покреће. Од те ширине ће зависити величина табле, умањена за 50 како бисмо добили дистанцу од ивице екрана. На основу ширине добијамо и величину појединачне ћелије.

```
@Override
protected void onMeasure(int width, int height){
    super.onMeasure(width, height);

    int dimenzija = Math.min(this.getMeasuredWidth(), this.getMeasuredHeight()) - 50;
    dimenzijaCeliye = dimenzija/9;

    setMeasuredDimension(dimenzija, dimenzija);
}
```

Слика 4.1.1 : Функција за добијање
ширине екрана

Рedefинисањем OnDraw методе постављамо почетне параметре за исцртавање табле које ћемо касније употребити у дизајнирању самих елемената.

```
@Override
protected void onDraw(Canvas canvas){
    bojaTableFarba.setStyle(Paint.Style.STROKE);
    bojaTableFarba.setStrokeWidth(16);
    bojaTableFarba.setColor(bojaTable);
    bojaTableFarba.setAntiAlias(true);

    bojaPopunjeneCelijeFarba.setStyle(Paint.Style.FILL);
    bojaPopunjeneCelijeFarba.setAntiAlias(true);
    bojaPopunjeneCelijeFarba.setColor(bojaPopunjeneCelije);

    bojaOznaceneKolRedFarba.setStyle(Paint.Style.FILL);
    bojaOznaceneKolRedFarba.setAntiAlias(true);
    bojaOznaceneKolRedFarba.setColor(bojaOznaceneKolRed);

    bojaBrojaKorisnikFarba.setStyle(Paint.Style.FILL);
    bojaBrojaKorisnikFarba.setAntiAlias(true);
    bojaBrojaKorisnikFarba.setColor(bojaBrojaKorisnik);

    bojaCelije(canvas, resavac.getOznacenRed(), resavac.getOznacenaKol());
    canvas.drawRect( left: 0, top: 0,getWidth(), getHeight(), bojaTableFarba);
    crtanjeTable(canvas);

    crtajBrojeve(canvas);
}
```

Слика 4.1.2 : Дефинисање почетних параметара

```
<com.example.sudoku.Tabla
    android:id="@+id/sudokuTabla"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    custom:bojaOznaceneKolRed="#f5f5f5"
    custom:bojaPopunjeneCelije="#cfcfcf"
    custom:bojaTable="#000000"
    custom:bojaBrojaKorisnik="#000000"
    custom:bojaBrojaBot="#029fe5"
    custom:bojaZaSakrivanje="@color/white"
    custom:layout_constraintBottom_toBottomOf="parent"
    custom:layout_constraintEnd_toEndOf="parent"
    custom:layout_constraintHorizontal_bias="0.473"
    custom:layout_constraintStart_toStartOf="parent"
    custom:layout_constraintTop_toTopOf="parent"
    custom:layout_constraintVertical_bias="0.283">

</com.example.sudoku.Tabla>
```

Слика 4.1.3 : Имплементација погледа

Функција која исцртава бројеве на саму таблу функционише преко угњеждене фор петрље и позиционих координата:

```
for(int r=0;r<9;r++){
    for(int k=0;k<9;k++){
        if(resavac.getTabla()[r][k] != 0){
            String tekst;
            tekst = Integer.toString(resavac.getTabla()[r][k]);
            float width, height;

            bojaBrojaKorisnikFarba.getTextBounds(tekst, start: 0, tekst.length(), bojaBrojaKorisnikOkvir);
            width = bojaBrojaKorisnikFarba.measureText(tekst) ;
            height = bojaBrojaKorisnikOkvir.height();

            canvas.drawText(tekst, x: (k*dimenzijaCeliJe) + ((dimenzijaCeliJe - width)/2),
                            y: (r*dimenzijaCeliJe+dimenzijaCeliJe) - ((dimenzijaCeliJe - height)/2),
                            bojaBrojaKorisnikFarba);
        }
    }
}
```

Слика 4.1 : Функција за
исцртавање бројева

Да бисмо добили координате ћелије на коју је корисник кликнуо прегазићемо onTouch методу на следећи начин.[2]

```
@SuppressWarnings("ClickableViewAccessibility")
@Override
public boolean onTouchEvent(@NonNull MotionEvent event){
    boolean isValid;

    float x = event.getX();
    float y = event.getY();

    int action = event.getAction();

    if (action == MotionEvent.ACTION_DOWN){
        resavac.setOznacenaKol((int) Math.ceil(x/dimenzijaCeliJe));
        resavac.setOznacenRed((int) Math.ceil(y/dimenzijaCeliJe));
        isValid = true;
    }
    else{
        isValid=false;
    }

    return isValid;
}
```

Слика 4.2.1 : добијање координата
притиснуте ћелије

Функција која исцртава таблу функционише преко две фор петље и позиционих координата које добијамо преко системских функција:

```
private void crtanjeTable(Canvas canvas){

    for(int c=0; c<10; c++){
        if(c%3==0){
            crtanjeDebeleLinije();
        }
        else{
            crtanjeTankeLinije();
        }
        canvas.drawLine( startX: dimenzijaCelije*c, startY: 0, stopX: dimenzijaCelije*c, getWidth(), bojaTableFarba);
    }
    for(int r=0; r<10; r++){
        if(r%3==0){
            crtanjeDebeleLinije();
        }
        else{
            crtanjeTankeLinije();
        }
        canvas.drawLine( startX: 0, startY: dimenzijaCelije*r, getWidth(), stopY: dimenzijaCelije*r, bojaTableFarba);
    }
}
```

Слика 4.2 : Функција за цртање
табле

Помоћу методе бојаЋелије реализујемо истицање реда и колоне у којој се обележена ћелија налази и тако обезбеђујемо кориснику лакшу оријентацију кроз апликацију.

```
private void bojaCelije(Canvas canvas, int red, int kol){
    if(resavac.getOznacenaKol() != -1 && resavac.getOznacenRed() != -1){
        canvas.drawRect( left: (kol-1)*dimenzijaCelije, top: 0, right: kol*dimenzijaCelije,
            bottom: dimenzijaCelije*9, bojaOznaceneKolRedFarba);
        canvas.drawRect( left: 0, top: (red-1)*dimenzijaCelije, right: dimenzijaCelije*9,
            bottom: red*dimenzijaCelije, bojaOznaceneKolRedFarba);
        canvas.drawRect( left: (kol-1)*dimenzijaCelije, top: (red-1)*dimenzijaCelije,
            right: kol*dimenzijaCelije, bottom: red*dimenzijaCelije, bojaPopunjeneCelijeFarba);
    }

    invalidate();
}
```

Слика 4.2.2 :
Функција за
истицање реда и
колоне

Activity_main.xml

Помоћу xml датотеке се бавимо дизајном саме апликације. Да бисмо реализовали само таблу направили смо персонализованани поглед а траку са улазним бројевима реализовали смо помоћу линеарног ређања дугмади.

```
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="457dp"
    android:layout_height="50dp"
    android:layout_marginTop="1dp"
    android:orientation="horizontal"
    custom:layout_constraintStart_toStartOf="parent"
    custom:layout_constraintTop_toBottomOf="@+id/sudokuTabla">

    <Button
        android:id="@+id/button"
        style="@style/Widget.AppCompat.Button.Borderless"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="DugmeJedan"
        android:text="1"
        android:textColor="#aaa69e"
        android:textSize="20sp" />
```

Слика 4.2.3 : Део кода реализације
траке са цифрама за унос

LogikaResenja.java

Решавање саме пузле се врши помоћу рекурзије. Након што иницијализујемо сва празна поља на нулу, уместо нуле уписујемо први број који је могуће да буде на тој позицији и такву таблу шаљемо у исту функцију. Уколико у било ком тренутку наиђемо на нелогичност уместо цифре коју смо првобитно унели уписујемо прву следећу цифру која је могуће бити на том пољу све док не решимо целу пузлу. [3]

```
for (int i=1; i<10; i++){
    this.tabla[row][col] = i;
    display.invalidate();

    if (proveri(row, col)){
        if (resi(display)){
            return true;
        }
    }

    this.tabla[row][col] = 0;
}
```

Слика 4.3 Функција за решавање
Судоку пузле

Део кода који проверава да ли је уписана цифра могућа у датој ћелији изгледа овако:

```
private boolean proverj(int row, int col){
    if (this.tabla[row][col] > 0){
        for (int i=0; i<9; i++){
            if (this.tabla[i][col] == this.tabla[row][col] && row != i){
                return false;
            }

            if (this.tabla[row][i] == this.tabla[row][col] && col != i){
                return false;
            }
        }
    }

    int boxRow = row/3*3;
    int boxCol = col/3*3;

    for (int r=boxRow; r<boxRow + 3; r++){
        for (int c=boxCol; c<boxCol + 3; c++){
            if (this.tabla[r][c] == this.tabla[row][col] && row != r && col != c){
                return false;
            }
        }
    }

    return true;
}
```

Слика 4.3.2 Функција за
проверавање Судоку пазле

Функција за уписивање броја у обележену ћелију је реализована на следећи начин:

```
public void setPozicijaBroja(int broj){
    if(this.oznacenaKol != -1 && this.oznaceniRed != -1){
        if(this.tabla[this.oznaceniRed-1][this.oznacenaKol-1] == broj){
            this.tabla[this.oznaceniRed-1][this.oznacenaKol-1] = 0;
        }
        else{
            this.tabla[this.oznaceniRed-1][this.oznacenaKol-1] = broj;
        }
    }
}
```

Слика 4.3.3 Функција за уписивање
броја у таблу

Подразумевни индекси су -1 како бисмо изоставили могућност уписа уколико корисник није обележио ћелију у коју жели да унесе број.

Апликација користи само библиотеке које су већ уграђене у сам Андроид студио па нема потребе да увозимо додатне библиотеке.

```
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Rect;
import android.util.AttributeSet;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import java.util.ArrayList;
import java.util.Random;
```

Слика 4.4

Хардверска реализација система

За покретање апликације је довољан било који паметан телефон са Андроид оперативним системом изнад верзије 6.0.[4]

Сама апликација није хардверски захтевна тако да ће сви уређаји функционисати и користити је без икаквих проблема.

Литература

1. <https://srbodroid.com/> - феб. 2023.
2. <https://github.com/bizz84/SudokuSolver> - дец. 2022.
3. https://www.youtube.com/watch?v=G_UYXzGuqvM – дец. 2022.
4. Андроид студио – дец. 2022.

Структура документације рађена по примеру РМА – dokumentacija
(moodle.fink.rs/mod/resource/view.php?id=15854)