



DogTime

אפליקציה שאוהבת כלבים



מגישה: גל רואש

בית ספר: מדעים, רחובות

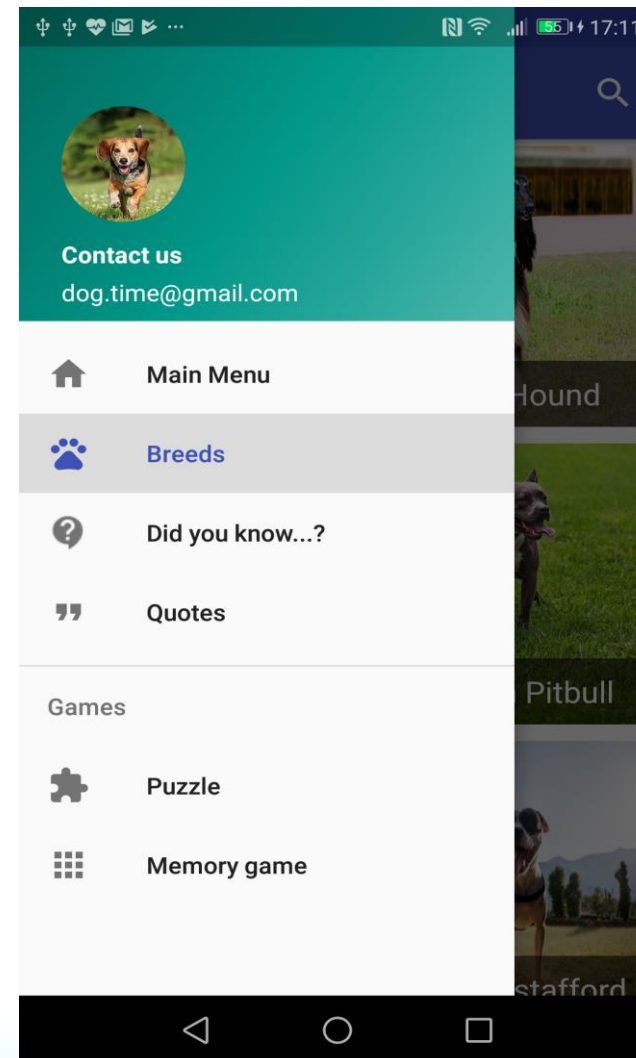
ת.ז: 211680749

מורה מגישה: זהבה לביא

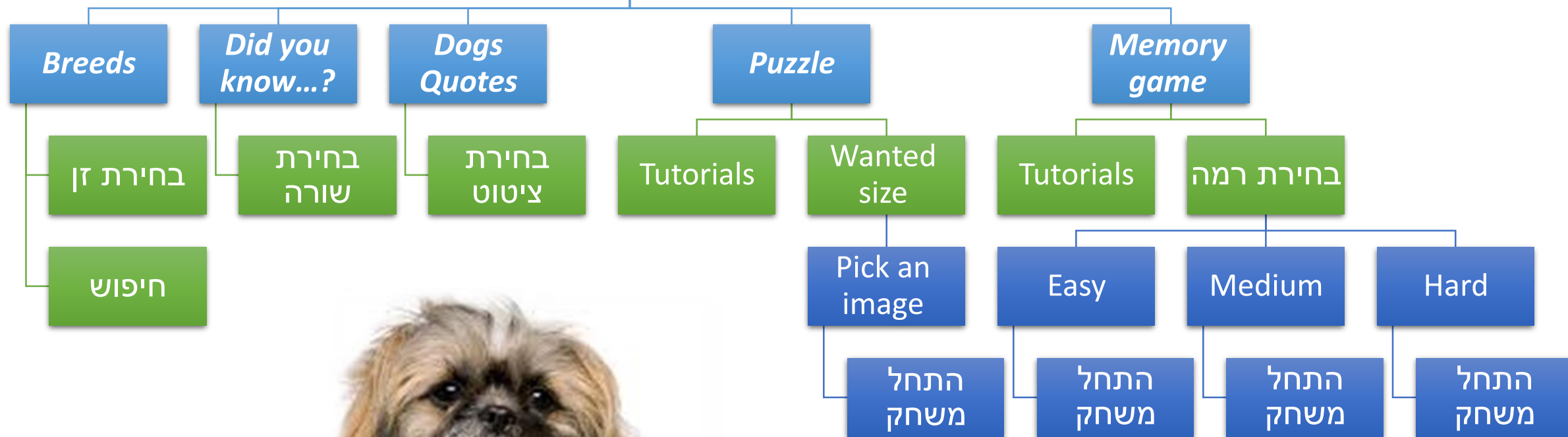
מסך הפתיחה של האפליקציה

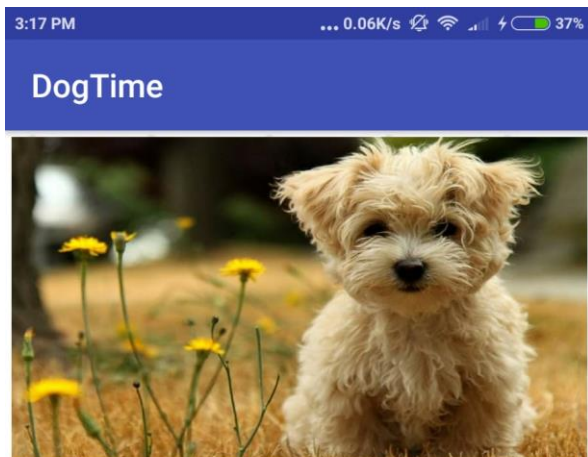
במסך הפתיחה יש 5 אייקונים, כאשר
כל אחד מהם מוביל למסך אחר.

ניתן להגיע למסכים האחרים גם על
ידי התפריט הנשלף.



מסך ראשי





Affenpinscher

Height

9 inches to 11 inches tall at the shoulder

Weight

7 to 9 pounds

Life Span



מסך

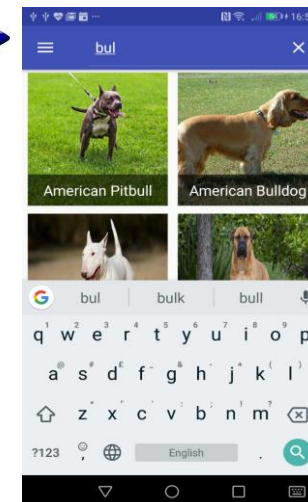
באפליקציה מוצע מידע אינפורמטיבי
רחב על זני כלבים – גובה, משקל,
תוחלת חיים, מאפיינים אישיותיים,
מידע בריאותי, האכלה ועוד

[להסבר על הקוד](#)



למידע נוסף, ניתן לגלול את המסך

לחיצה על הזכוכית המגדלת,
מאפשרת חיפוש זן על ידי הקלדת
אותיות המוכלות בשמו.





מסך

בכניסה לאחת השורות, מתקבל מידע המתאים לכוותרת בשורה, בו ניתן לגלגל את המסך ולהתקדם בו. כמו כן, ניתן גם להקטין ולהגדיל את התמונה הנבחרת בעזרת מגע במסך.

להסבר על הקוד



לנושאים נוספים, ניתן לגלול את המסך



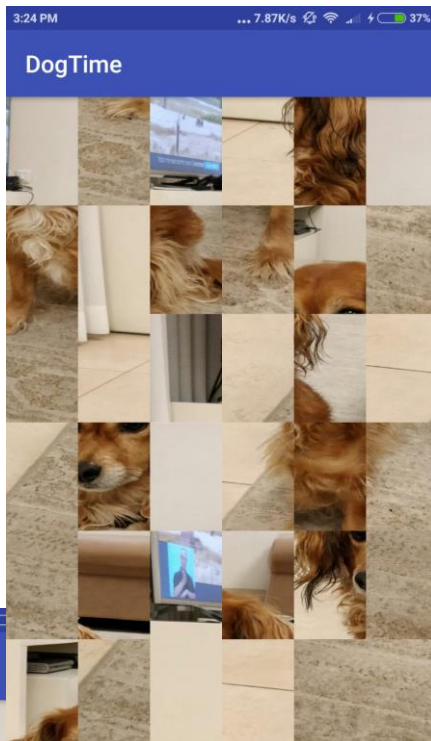
תוספת מחויכת המוצעת באפליקציה היא מגוון
ציטוטים משעשעים ומרגשים על כלבים.
הציטוטים מופיעים על תמונות כלבים.
ניתן לעמוד על אחד הציטוטים ולראות אותו
בהגדלה. ניתן גם להקטין ולהגדיל את התמונה
הנבחרת בעזרת מגע במסך.

[להסבר על הקוד](#)



לציטוטים נוספים, ניתן לגלול את המסך

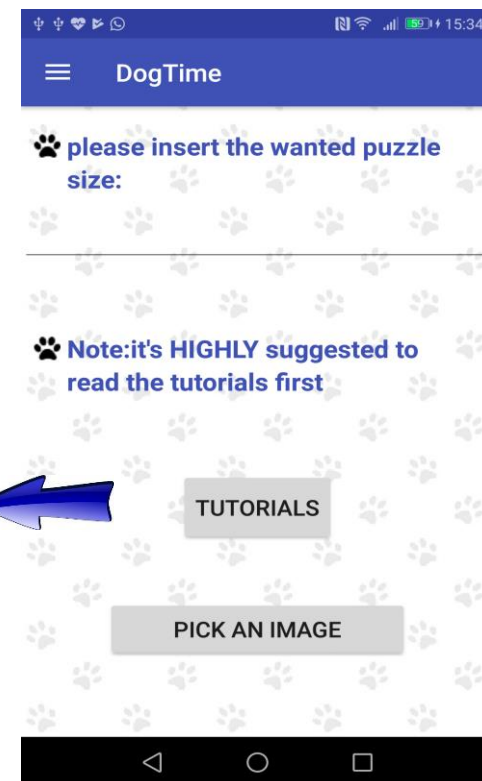




מסר

באפליקציה מוצע משחק פאזל.
המשתמש בוחר תמונה מהגלריה
שלו, או מצלם תמונה ואז בוחר
מספר חלקים מבוקש בפאזל.
האפליקציה מפצלת את התמונה
למספר החלקים המבוקש,
ומערבבת ביניהם באופן רנדומלי.
הפאזל הוא מסוג "sliding puzzle".
המשחק מלווה במוזיקת רקע.

[להסבר על הקוד](#)





מסך

משחק זיכרון, עם תמונות כלבים.
בתפריט הראשי של המשחק ניתן

לבחור רמת קושי:

קל (6 תמונות),

בינוני (12 תמונות)

או קשה (16 תמונות).

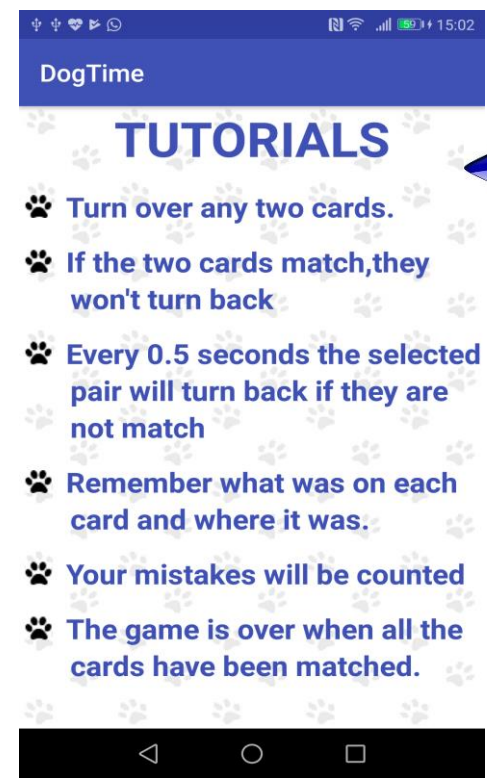
במהלך המשחק, האפליקציה מונה

את מספר הטעויות שהמשתמש

טועה, עד לסיומו.

המשחק מלווה במוזיקת רקע.

[להסבר על הקוד](#)





קוד

```
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        View view;
        Dog dog = myBreedsList.get(position);
        if (convertView == null) { // if it's not recycled, initialize some attributes
            // I inflate the xml which will give me a view
            LayoutInflater inflater = (LayoutInflater) myContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            convertView = inflater.inflate(R.layout.dog_list_layout, parent, attachToRoot: false);
            ImageView ivdog = (ImageView) convertView.findViewById(R.id.ivdog);
            TextView tvName = (TextView) convertView.findViewById(R.id.breedname);

            final ViewHolder viewHolder = new ViewHolder(tvName, ivdog);
            convertView.setTag(viewHolder);
        }
        final ViewHolder viewHolder = (ViewHolder) convertView.getTag();

        viewHolder.tvName.setText(myBreedsList.get(position).getBreedname());

        int drawableId = 0; //init drawable
        try {
            Class res = R.drawable.class;
            Field field = res.getField(myBreedsList.get(position).getImage().toString());
            Log.e(tag: "kk", myBreedsList.get(position).getImage().toString());
            drawableId = field.getInt(obj: null);
        } catch (Exception e) {
            Log.e(tag: "MyTag", msg: "Failure to get drawable id.", e);
        }

        Glide.with(myContext).load(drawableId).into(viewHolder.ivdog); //using glide library to load pictures faster
        return convertView;
    }
```

כל זני הכלבים אשר מוצגים על המסך, נקראו ממסד הנתונים שלי והושמו לתוך רשימה מסוג Dog.

כדי לעצב דף זה השתמשתי ב

GridView.

השימוש ב GridView הוא מורכב

וכולל כתיבת עמוד xml, בנוסף,

בניית מחלקת Adapter, שהיא זו

שבונה את ה GridView.

נעשה שימוש בספריית glide.

```
Glide.with(myContext).load(drawableId).into(viewHolder.ivdog);
```



קוד

עיצוב תא אחד

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.gal.dogtime.BreedsActivity">

    <!--the xml file of breedsctivity-->
    <!-- TODO: Update blank fragment layout -->

    <GridView
        android:paddingTop="60dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/gridViewDogs"
        android:numColumns="2"
        android:scrollbars="horizontal"
    />

</LinearLayout>
```

```
<FrameLayout android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <ImageView
        android:id="@+id/ivdog"
        android:layout_width="185dp"
        android:layout_height="180dp"
        android:paddingBottom="10dp"
        android:paddingLeft="7dp"
        android:paddingRight="10dp"
        app:srcCompat="@mipmap/ic_launcher"
        android:scaleType="fitXY" />

    <TextView
        android:id="@+id/breedname"
        android:layout_width="168dp"
        android:layout_height="40dp"
        android:layout_marginLeft="7dp"
        android:layout_marginRight="11dp"
        android:gravity="center"
        android:layout_marginTop="130dp"
        android:cursorVisible="true"
        android:textSize="18dp"
        android:textColor="#ffffff"
        android:background="#99000000" />

</FrameLayout>
```



קוד

InfoBreedsActivity

```
myBreedsList= (ArrayList<Dog>) intent.getSerializableExtra( name: "myBreedsList");  
  
//here im setting texts into the views  
name.setText(myBreedsList.get(id-1).getBreedname());  
breedHeight.setText(myBreedsList.get(id-1).getHeight());  
breedWeight.setText(myBreedsList.get(id-1).getWeight());  
lifeSpan.setText(myBreedsList.get(id-1).getLifeSpan());  
personality.setText(myBreedsList.get(id-1).getPersonality());  
health.setText(myBreedsList.get(id-1).getHealth());  
feeding.setText(myBreedsList.get(id-1).getFeeding());  
childrenAndPets.setText(myBreedsList.get(id-1).getChildrenAndPets());  
care.setText(myBreedsList.get(id-1).getCare());
```

כאשר המשתמש לוחץ על אחד מזני הכלבים הוא מגיע לדף אשר מציג מידע רחב ומפורט על זן זה.

עיקר העיצוב במסך שמציג את המידע על הזן נעשה ב־xml.

את המידע קראתי מהרשימה מסוג Dog, אשר הועברה מהמסך הקודם.





קוד

עיקר העיצוב במסך זה נעשה ב xml,

כל אחד מהתאים הוא ImageButton.

השתמשי ב ScrollView כדי

שהמשתמש יוכל לגלגל, והשתמשי

בספריית glide על מנת שהגלגול

יהיה מהיר וחלק.

כאשר המשתמש לוחץ על אחת

מהתמונות, הוא מגיע למסך אשר

מציג את המידע שנוגע לתמונה.

```
imgbtns[i].setOnClickListener((v, event) -> {  
    if(event.getAction() == MotionEvent.ACTION_UP){  
        int j=0;  
  
        for( int i=0;i<16;i++)  
        {  
            j++;  
  
            int id=getResources().getIdentifier( name: "iconinfo"+j, defType: "drawable",getContext().getPackageName()  
  
            if(v.getId()==iconsInfo.getChildAt(i).getId())//checking what was clicked  
            {  
                Intent intent=new Intent(getActivity(),DisplayFcatsActivity.class);  
                int position=j;  
                intent.putExtra( name: "position",position);  
                startActivity(intent);  
            }  
        }  
  
        return true;  
    }  
})
```



קוד

במסך זה עשיתי שימוש בספריית Zoomage, התמונה מוצגת בתוך view מסוג ZoomageView. ספריה זו למעשה מאפשרת למשתמש להגדיל ולהקטין את התמונה כרצונו.

```
ZoomageView ivfact=(ZoomageView) findViewById(R.id.ivfact);
```





כדי להציג את כל הציטוטים במסך זה באופן מיטבי השתמשתי GridView.
השימוש ב GridView הוא מורכב וכולל כתיבת עמוד xml, בנוסף, בניית מחלקת Adapter,
שהיא זו שבונה את ה GridView.

נעשה שימוש בספריית glide.





קוד

בחירת התמונה מתבצעת

כשהמשתמש לוחץ על כפתור

"PICK AN IMAGE" מופיע דיאלוג

ובו המשתמש בוחר את המקום

ממנו ירצה לקחת את התמונה –

גלריה או צילום התמונה במצלמה.

את התמונה הנבחרת אעביר למסך

הבא כ String.

```
private void selectImage()
{
    final CharSequence[] items={"Camera","Gallery","Cancel"};
    AlertDialog.Builder builder=new AlertDialog.Builder(getActivity());
    builder.setTitle("Add Image");

    builder.setItems(items, (dialog, i) → {
        if(items[i].equals("Camera")){
            dispatchTakePictureIntent();
        }
        else if(items[i].equals("Gallery")){
            Intent intent=new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            intent.setType("image/*");
            startActivityForResult(intent.createChooser(intent, title: "Select File"), SELECT_FILE);
        }
        else if(items[i].equals("Cancel")){
            dialog.dismiss();
        }
    });
    builder.show();
}
```



קוד

אם המשתמש בחר לקחת תמונה מהמצלמה

לקיחת תמונה מהמצלמה ושליחתה כ String,
זו פעולה מורכבת.
יצירת קובץ ריק בפורמט URI ושם אחסן את
התמונה שהמשתמש יקח.

אם המשתמש בחר לקחת תמונה מהגלריה

לקיחת תמונה מהגלריה, זו פעולה הדורשת
להגדיר את ה Intent, איזה סוג תמונות ילקחו,
ולהפעיל את onActivityResult עם הפרמטרים
המתאימים.
התמונה שתוחזר תהיה בפורמט URI ואותו
אמיר ל String.



קוד

```
private void scramble() {  
    int index;  
    int temp;  
    Random random = new Random();  
    //here im actually doing the switching  
    for (int i = tileList.length - 1; i > 0; i--) {  
        index = random.nextInt(bound: i + 1); //picks a random number in the bound of i+1  
        temp = tileList[index]; //i put in temp the value of the tileList in position of index  
        tileList[index] = tileList[i]; //here i put the value in position i into the value in poistion index  
        tileList[i] = temp;  
    }  
}
```

לאחר שהשתמש הזין את גודל
הפאזל הרצוי, ובחר תמונה מהגלריה
או צילם תמונה במצלמה, התמונה
הנבחרת נחלקת למספר חלקי הפאזל
וחלקי הפאזל נכנסים לרשימה
drawableImages.
כדי לערבב את החלקים בצורה
היעילה ביותר, יצרתי מערך ששמו
tileList ובו למעשה סדר האינדקסים
כסדר התמונות. את המערך הזה אני
מערבבת באופן רנדומלי.



קוד

הצגת הפאזל

גובה ורוחב חלקי הפאזל משתנים כשמספר חלקי הפאזל משתנים.

כדי לעשות את הפאזל דינמי, אני מחשבת בפונקציה `setDimensions()` את הגובה והרוחב שכל חלק צריך להיות.



קוד

מהלך המשחק

יצרתי מחלקה ששמה `GesturDetectGridView` אשר תפקידה העיקרי הוא לבדוק אם המגע היה תקין או לא, האם הוא למעלה, למטה, ימינה או שמאלה. אם המשתמש החליק את האצבע כלפי מטה, מחלקת `GesturDetectGridView` תפעיל את הפונקציה `moveTiles` ותשלח את הפרמטרים המתאימים כדי שהפונקציה תפעל בהתאם.





קוד

כדי לכתוב את משחק הזיכרון כתבתי מחלקה ושמה Card, המייצגת כל אחד מן הקלפים שלי.
במחלקה זו למעשה יש שלושה מערכים:

המערך הראשון הוא מערך של אינדקסים של תמונות, וגודלו מחצית ממספר הקלפים (למשל אם במשחק יש 6 קלפים גודל המערך יהיה 3).

המערך השני הוא מערך של קלפים אשר כל תא בו הוא מסוג Card וגודלו כגודל מספר הקלפים.

המערך השלישי הוא מערך של ImageViews שכל תא הוא מסוג זה וגודלו הוא כגודל הקלפים.

```
int numberOfClicks=0;
int mistakes=0;
TextView numOfMistakes;//textView that holds the number of mistakes
public static int counter=0;//so i can know when the games over
Card previousCard=new Card();//the previous card cuz i need to find a match
final int ARR_SIZE = 6;//the number of cards is my ARR_SIZE
Card[] cards=new Card[ARR_SIZE];//an array of cards
ImageView[] views = new ImageView[ARR_SIZE];//array of views
int[] dogs=new int[ARR_SIZE/2];//the size of my dog array(an array for the images index) is half of the ARR_SIZE
```




קוד

בשלב ראשון האלגוריתם מערבב את הקלפים על ידי שימוש בפעולה `pickrandom()`, בשלב השני מקבלת את הבחירה של המשתמש ובודקת האם הקלף השני שהוא לחץ עליו תואם או לא, אם זה לא תואם אז שני הקלפים יתהפכו חזרה תוך 0.5 שניות, אחרת, הם יישארו חשופים.

כמו כן, יש שימוש בservice במסך זה - כאשר המשתמש משחק ישנה מוזיקת רקע שמתנגנת.





תודה על ההקשבה