תיק פרויקט

15 puzzle

מגיש: ישראל שראל

ת.ז: 214762494

בית הספר: ישיבת בני עקיבא השומרון

שם המנחה: דבורה יעיש

שם החלופה: טלפונים ניידים

תוכן עניינים

מבוא:

הרקע לפרויקט:

שם הפרויקט הוא ,15 puzzle זהו משחק ישן שבנוי על לוח עם אריחים ומטרתו היא לסדר את המספרים לפי הסדר מ1 עד המספר הכי גדול.

**כללי המשחק:** המשחק מתחיל כשהלוח מבולגן והמספרים לא במקום, בנוסף יש מקום אחד ריק וניתן להזיז רק אליו אריחים, המטרה היא לסדר את המספרים מ1 ועד המספר הכי גדול לפי הגודל שנקבע(הdefult הוא 4x4).

קהל היעד של המשחק הוא כל אחד שרוצה לשחק במשחק בזמנו החופשי ולהפעיל את המוח.

הסיבה שבחרתי את הנושא הנל זה שעניין אותי לבנות משהו שלא רק מצריך ידע במושגים בתכנות ובכל מיני דברים טכניים אלא גם חשיבה מעמיקה ואלגוריתמיקה ובמשחקים כמו זה יש הרבה דברים שצריך לחשוב לגבם ולהפעיל את המוח.

יכולות המערכת:

* לשחק במשחק
* לערבב את המשחק ולהתחיל מהתחלה בכל שלב
* לעצור את המשחק ולהמשיך
* לשנות את הגודל של הלוח וכך גם את הצבע של האריחים
* לראות את השיאים מסודרים לפי מהלכים ולפי הזמן
* לשלוח הודעת sms למפתח.

תהליך המחקר:

בהתחלה לקחתי את המשחק ושיחקתי איתו קצת וניסיתי לחשוב איזה קשיים יכולים להיות בבניית המשחק, לאחר מכן הסתכלתי על הערך שלו בוויקיפדיה ולמדתי עוד קצת על האופי של המשחק.

לאחר מכן ניסיתי לחשוב באיזה צורה של תכנות אני אצטרך להשתמש בשביל לבנות את המשחק בצורה הכי אידיאלית ואלגנטית ולכן הסתכלתי קצת באתר github על פרויקטים דומים בשביל לקבל איזה נקודת מבט על איך אני רוצה שהפרויקט יהיה בנוי, שמה ראיתי שדרך מאוד טובה לעבוד עם משחקים כמו זה זה בניית class שיורש מview ועל זה לעבוד אז התחלתי קצת ללמוד על זה והתחלתי להתנסות עם זה.

כמובן שהסתכלתי גם על אפליקציות דומות וראיתי מה המודל הכי טוב בשביל הפרויקט הזה מכל מיני מובנים.

אתגרים מרכזיים:

בעיה רצינית שהיה לי זה איך לבנות את הלוח של המשחק בצורה הכי אלגנטית וקלה לשימוש ועוד בעיה שאיתה התמודדתי זה שיכול לצאת מצב שבו הלוח יהיה לא פתיר והייתי צריך למצוא אלגוריתם שפותר את הבעיה וגם התמודדתי עם המורכבות של לשלב בסיס נתונים וsql באפליקציה.

עוד אתגר מאוד רציני שנתקלתי בו זה המעבר בין מצב שהמשחק עוצר למצב שהוא רץ או מצב שהוא מתחיל מחדשת מכיוון שיש הרבה ואריאציות של מעברים היה לי קשה להצליח לתפעל את כולם בצורה הנכונה. בשביל להתמודד עם בעיה זו ממש הייתי צריך לתכנן ולבנות סוג של מודל חישובי שיקל עלי בהתייחסות לכל המצבים.

אלמנט טכנולוגי חדשני זה שהשתמשתי בrecycleView במקום בlistView ובשביל זה הייתי צריך ליבא ספרייה בgradle.

**תיאור תחום הידע – פרק מילולי**

אובייקטים:

בשביל משחק זה בניתי מספר אובייקטים:

Tile: אובייקט שמייצג אריח אחד עם מספר על הלוח.

Square: אובייקט שמייצג מקום בלוח שבו יכול להיכנס אריח(Tile).

BoardGame: אובייקט שיורש מView ומכיל מערך דו ממדי של Tile ומערך דו ממדי של Square.

Game: אובייקט ששולט בכל המשחק מבחינת המצב ריצה שלו או היצירה בכלל של הBoardGame ועוד.

Time: אובייקט שיורש מThread ואחראי על השעון שסופר את הזמן.

Record: אובייקט שמייצג יחידה אחת של שיא.

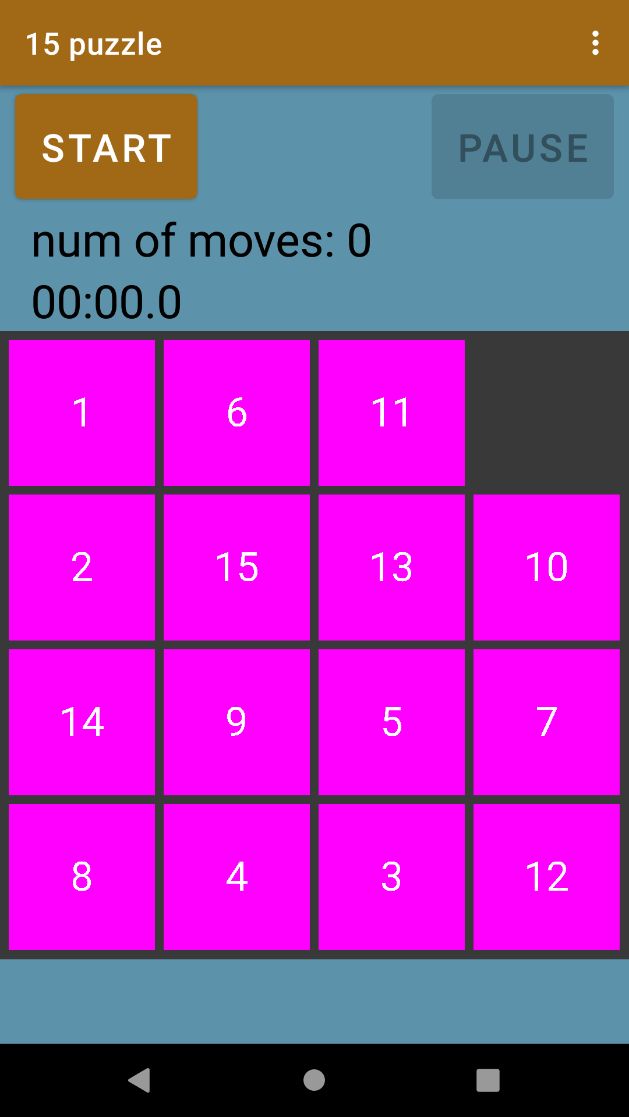
RecordAdapter: אובייקט שיורש מ RecyclerViewודואג לסדר את הרקורדים בסוג של רשימה.

RecordHelper: אובייקט שיורש מ SQLiteOpenHelperודואג לשמור את הנתונים בdataBase פנימי של האפליקציה.

כאמור השתמשתי בייצוג מידע של מערך דו ממדי בעיקר בשביל לייצג נתונים כמו האריחים או המקומות על הלוח(Square) או מערך של מספרים בשביל הסדר שלהם על הלוח וגם שימוש בטבלה(database) בשביל השיאים ויש לי פעולה של הוספה של שיא:createRecord(Record r) ושל קבלה של כל השיאים: getAllRecord()

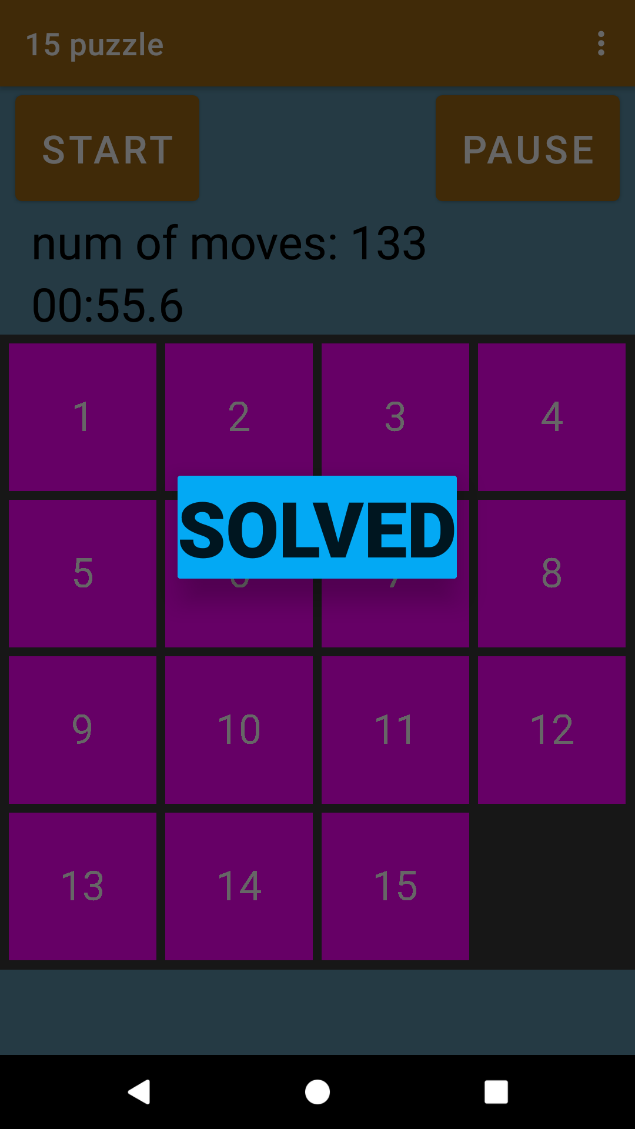
**מבנה / ארכיטקטורה של הפרויקט**

Main Activity – המסך הראשי



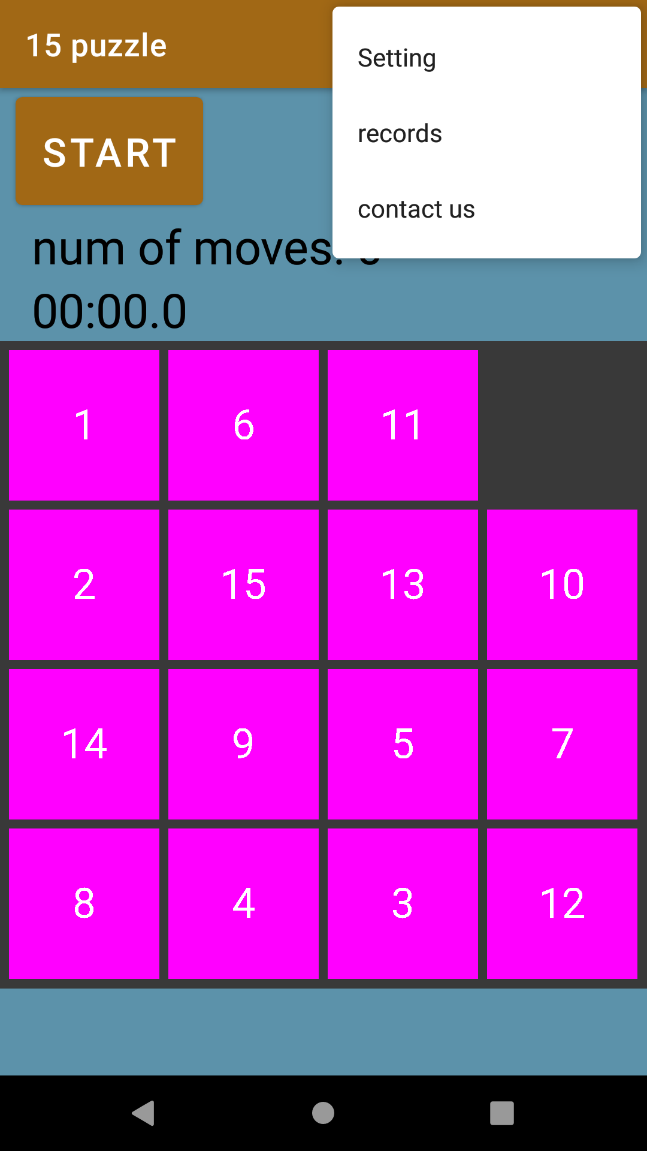
המסך הזה הוא המסך הראשי של המשחק ובו מוטמע האובייקט של הלוח BoardGame בLinearLayout. כשלוחצים על הלוח אז הזמן מתחיל לרוץ ונהיה אפשרי ללחוץ על הpause ואם לוחצים על אריח ליד המקום הריק האריח זז לשם והnum of mobes עולה ב1. אם לוחצים על הstart הזמן והמהלכים מתאפסים והלוח מתערבב. אם לוחצים על הpause הכיתוב משתנה לcontinue והזמן נעצר ולא אפשרי להמשיך את המשחק עד שלוחצים על הcontinue ואז המשחק ממשיך והכתובית משתנה חזרה לpause. בonPause המשחק נעצר ובonCreate המשחק נוצר. לחיצה על ה3 נקודות תפתח תפריט.

Record Dialog - דיאלוג שיאים



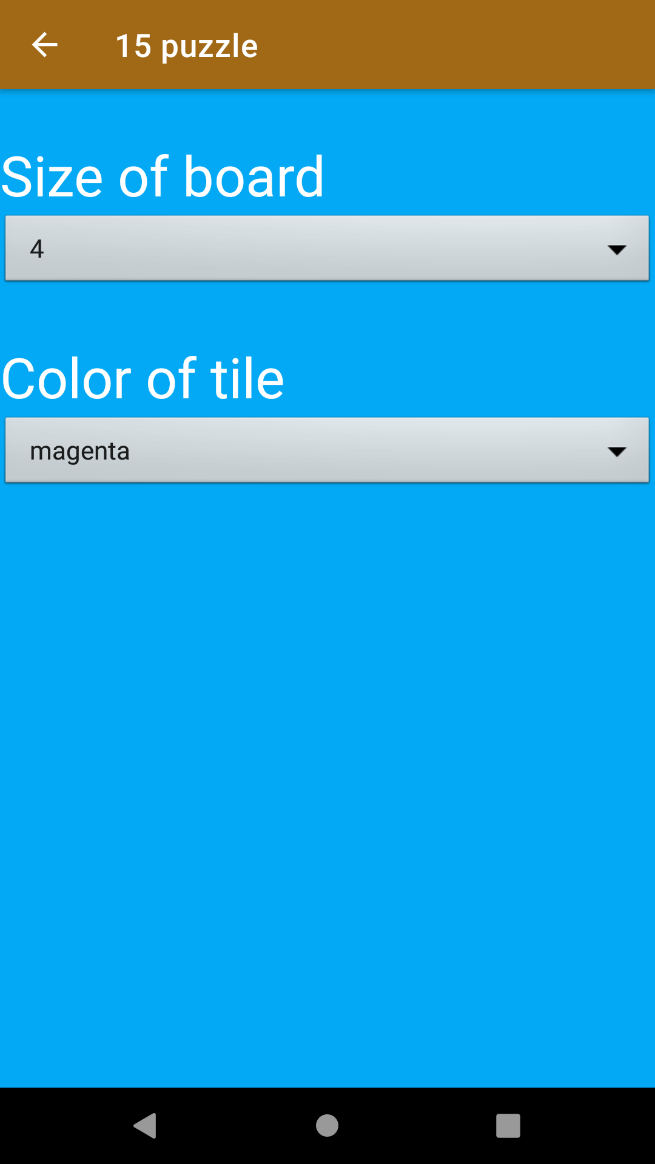
הדיאלוג הזה נפתח מתי שהשחקן פותר את הלוח וגומר את המשחק, המשחק נעצר והתוצאה שלו נשמרת בבסיס נתונים, בשביל לשחק שוב צריך ללחוץ על solved ואז זה מאפס את הזמן והמהלכים ומערבב את הלוח.

Menu main – תפריט ראשי



בתפריט הראשי יש שלושה אפשרויות

1. מעבר למסך הגדרות על ידי Intent(SettingActivity) ואיפוס המשחק.
2. פתיחה של dialog של השיאים לגודל הלוח הספיציפי שמוגדר ועצירת המשחק.
3. מעבר למסך של שליחת sms למפתח(SmsActivity) ואיפוס המשחק.

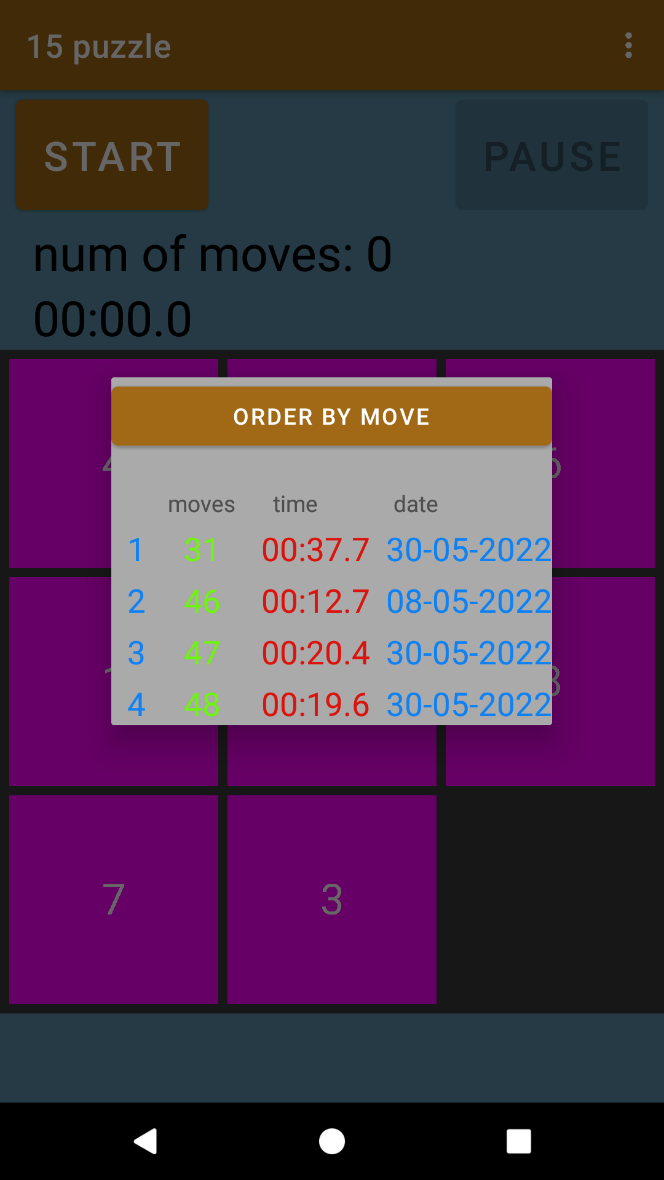
Setting Activity – מסך הגדרות

במסך זה ניתן לשנות את גודל הלוח של המשחק(Size of board) מגודל של 3x3 עד גודל של 10x10.

בנוסף ניתן לשנות את צבע האריחים לשלל צבעים כגון ירוק, אדום, סגול ועוד...

בלחיצה על החץ למעלה ניתן לחזור למסך הראשי(MainActivity) עם ההגדרות שנשמרו.

Record Dialog - דיאלוג שיאים



לאחר שחולצים על הכפתור record נפתח דיאלוג שמראה את כל השיאים שנשמרו בבסיס נתונים מסודרים בהתחלה לפי מספר מהלכים ואפשר לשנות שהשיאים יהיו מסודרים לפי הזמן ולאחר מכן לשנות חזרה, כשהדיאלוג נפתח המשחק נעצר, בשביל לצאת מהדיאלוג צריך ללחוץ על המסך סתם לא איפה שהדיאלוג נמצא.

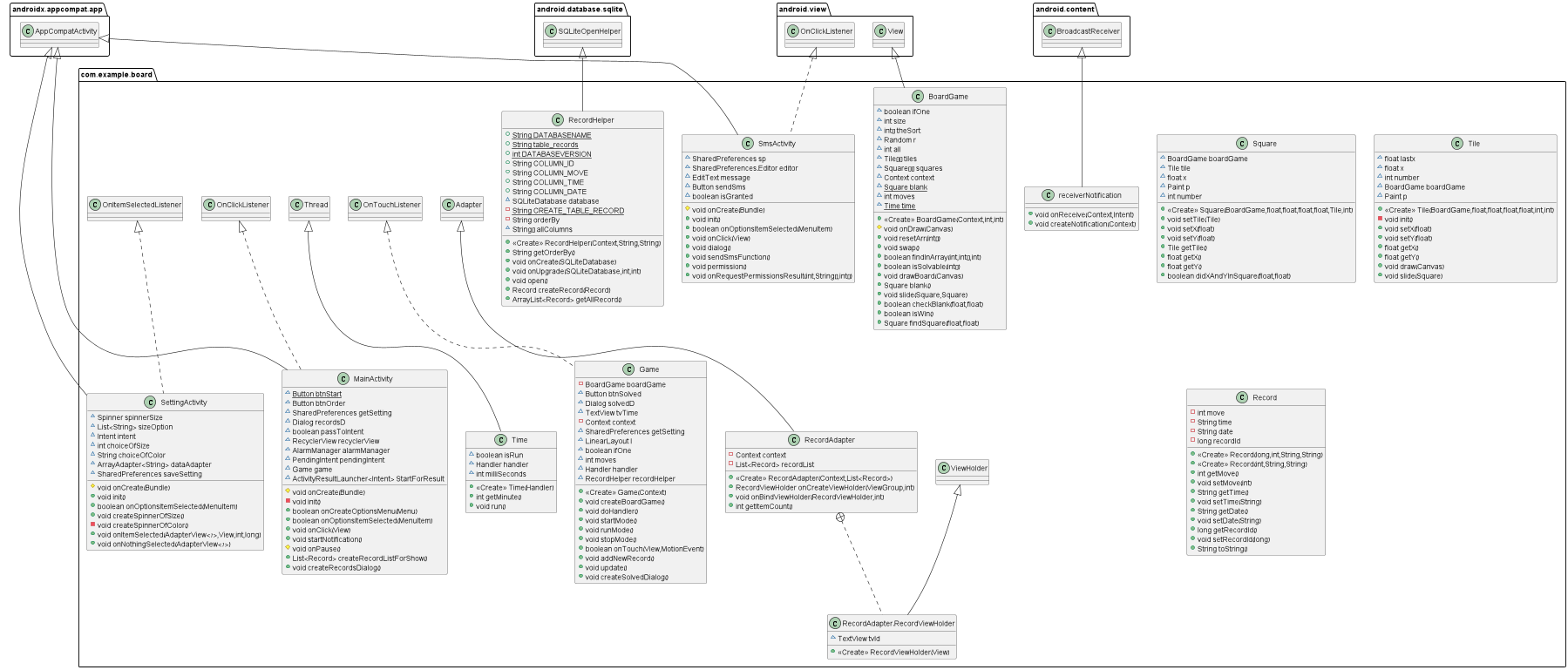
Sms Activity – מסך שליחת הודעת sms למפתח



במסך זה ניתן לכתוב הודעה למפתח ולשלוח לו sms, אם מנסים לשלוח הודעה ריקה מוקפץ toast שמסביר שאי אפשר לשלוח הודעה ריקה ובפעם הראשונה שמנסים לשלוח הודעת זה מפעיל בקשת permission לשליחת sms, בלחיצה על החץ למעלה ניתן לחזור למסך הראשי(MainActivity).

**Screen flow diagram**

**uml**



לתמונה באיכות יותר גבוהה:

<https://drive.google.com/file/d/10WRi1fPlqbmrfSLoh5dAOD8gRf2oHcT2/view?usp=sharing>

**מימוש הפרויקט**

חלק גרפי (קבצי תצורה/קונפיגורציה)

**activity\_main.xml:**

קובץ התצוגה הגרפית במסך הפתיחה(main activity) שבו מוצג הלוח של המשחק, בנוסף יש בו כפתור start שאחראי לערבב את הלוח ולאפס את הזמן והמהלכים וכפתור stop שעוצר את המשחק אם הוא רץ וממשיך אותו אם הוא בעצירה, בנוסף יש text view שמראה את מספר המהלכים ועוד text view שמראה את הזמן שרץ מאז שהתחלת לפתור.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#5C92AA"  
 tools:context=".MainActivity">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/btnStart"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="80dp"  
 android:layout\_alignParentLeft="true"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:text="start"  
 android:textSize="25sp"/>  
  
 <Button  
 android:id="@+id/btnPause"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="80dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_alignParentRight="true"  
 android:text="pause"  
 android:enabled="false"  
 android:textSize="25sp" />  
  
 </RelativeLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/tvMoves"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="right"  
 android:layout\_marginLeft="20dp"  
 android:text="number of moves: 0"  
 android:textColor="@color/black"  
 android:textSize="30sp" />  
  
 <TextView  
 android:id="@+id/tvTime"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="20dp"  
 android:layout\_gravity="left"  
 android:text="00:00.0"  
 android:textColor="@color/black"  
 android:textSize="30sp" />  
 </LinearLayout>  
   
 </LinearLayout>  
 <!--בתוך הLinearLayout מכניסים את הלוח של המשחק-->  
 <LinearLayout  
 android:layout\_gravity="center"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/lGame"  
 android:orientation="vertical" >  
 </LinearLayout>  
</LinearLayout>

**activity\_setting.xml:**

קובץ התצוגה הגרפית במסך ההגדרות(setting activity) שבו מוצג הגדרות של המשחק: text view שכתוב עליו size of board ומתחתיו spinner שאתה יכול לבחור בו איזה גודל של הלוח אתה רוצה ואותו דבר עם הגדרה של הצבע של האריחים, text view וspinner.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#03A9F4"  
 tools:context=".SettingActivity">  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="Size of board"  
 android:textSize="35sp"  
 android:textColor="#fff"/>  
  
 <Spinner  
 android:id="@+id/spinnerSize"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:background="@android:drawable/btn\_dropdown"  
 android:spinnerMode="dropdown"  
 android:prompt="@string/spinner\_title"  
 />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="Color of tile"  
 android:textSize="35sp"  
 android:textColor="#fff"/>  
  
 <Spinner  
 android:id="@+id/spinnerColor"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:background="@android:drawable/btn\_dropdown"  
 android:spinnerMode="dropdown"  
 android:prompt="@string/spinner\_title"  
 />  
</LinearLayout>

**activity\_sms.xml:**

קובץ התצוגה הגרפית במסך השליחת sms(sms activity) שבו מוצג edit text שבו המשתמש כותב את ההודעה וbutton בשביל לשלוח את ההודעת sms למפתח.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".SmsActivity">  
   
 <EditText  
 android:id="@+id/etSms"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:hint="your message here"  
 android:ems="10"  
 android:inputType="textMultiLine" />  
   
 <Button  
 android:id="@+id/btnSms"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Send SMS" />  
  
</LinearLayout>

**custom\_dialog\_records.xml:**

קובץ התצוגה הגרפית שמיועד להציג את הדיאלוג שמציג את השיאים, הוא מכיל כפתור שאחראי לשנות את הסידור של השיאים, table row שבתוכו כתוב move, time, date שמתחתם מוצג הדברים שהם מתארים והrecycle view שבתוכו השיאים שנשמרו בבסיס נתונים.

<?xml version="1.0" encoding="utf-8"?>  
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:background="#aaa"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/btnOrder"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="order by move"  
 android:layout\_marginBottom="20sp"/>  
 <TableRow  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="35dp"  
 >  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="moves"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="2"  
 android:text="time"/>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_weight="3"  
 android:text="date"/>  
   
 </TableRow>  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
</TableLayout>

**custom\_solved.xml:**

קובץ התצוגה הגרפית שמיועד להציג דיאלוג שכתוב עליו solved שקופץ מתי שהשחקן פותר את המשחק.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <android.widget.Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#03A9F4"  
 android:text="solved"  
 android:textSize="50sp"  
 android:textStyle="bold"  
 android:id="@+id/btnSolved"  
 />  
  
  
</LinearLayout>

**records\_layout.xml:**

קובץ התצוגה הגרפית שמיועד להציג פריסה של Record אחד שבו יש ארבע text view שמייצגים תאריך, זמן, מהלכים וid, אחר כך מממשים אותו בrecycle view לכל שיא ושיא.

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal">  
  
  
 <TextView  
 android:id="@+id/tvId"  
 android:layout\_width="25dp"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="5dp"  
 android:text="00"  
 android:textColor="#0e82f5"  
 android:textSize="20dp" />  
  
 <TextView  
 android:id="@+id/tvMoves"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
  
 android:layout\_marginTop="5dp"  
 android:text="000"  
 android:textColor="#6ef50e"  
 android:textSize="20dp" />  
  
 <TextView  
  
 android:id="@+id/tvTime"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="25dp"  
 android:layout\_marginTop="5dp"  
 android:text="00:00.0"  
 android:textColor="#dc130c"  
 android:textSize="20sp" />  
  
 <TextView  
 android:id="@+id/tvDate"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="10dp"  
  
 android:layout\_marginTop="5dp"  
 android:text="10.02.2022"  
 android:textColor="#0e82f5"  
 android:textSize="20dp" />  
  
</LinearLayout>

**menu\_main.xml:**

קובץ התצוגה הגרפית שמציג את התפריט הראשי שמוצג במסך הפתיחה(main activiry), יש בו item שמוביל למסך הגדרות(setting activity),item שמקפיץ את הדיאלוג של השיאים וitem שמוביל למסך לשליחת sms activity)sms).

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <item android:id="@+id/setting"  
 android:icon="@android:drawable/ic\_menu\_manage"  
 android:title="Setting"  
 app:showAsAction="never"/>  
  
 <item android:id="@+id/records"  
 android:icon="@android:drawable/ic\_menu\_sort\_by\_size"  
 android:title="records"  
 app:showAsAction="never"/>  
  
 <item android:id="@+id/contact"  
 android:icon="@android:drawable/ic\_menu\_sort\_by\_size"  
 android:title="contact us"  
 app:showAsAction="never"/>  
  
</menu>

**strings.xml:**

קובץ xml שאחראי לשמור מחרוזות במקום לכתוב אותם בקוד.

<resources>  
 <string name="app\_name">15 puzzle</string>  
 <string name="spinner\_title">spinner title</string>  
 <string name="message1">Don\'t forget to play today !!</string>  
 <string name="message2">Playing the game will improve your mind !</string>  
 <string name="message3">You still did not play today !? Go ahead !!</string>  
 <string name="message4">sad?? Bored?? This game will improve your mood instantly !!!</string>  
 <string name="message5">Come on! Take a minute and enjoy</string>  
</resources>

חלק לוג (קבצי קוד Java)

קבצי קוד - Activities

**MainActiviry:**

מחלקה ראשית שמתחברת לקובץ xml – activity\_main.xml והיא המסך הראשון שנכנסים אליו באפליקציה והוא המסך הראשי ובו נמצא המשחק ומשמה מנווטים לשאר האקטיביטים.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| static Button btnStart | כפתור שמאפס את המשחק ומערבב אותו. |
| static Button btnPause | כפתור שעוצר את המשחק אם הוא רץ וממשיך אותו אם הוא בעצירה. |
| Button btnOrder | כפתור בדיאלוג של השיאים שהוא מסדר את השיאים לפי זמן או מהלכים. |
| Dialog recordsD | הדיאלוג של השיאים. |
| boolean passToIntent | משתנה שבודק אם המשתנה עבר לאקטיביטי אחר עם intent. |
| RecyclerView recyclerView | סוג של view שמציג את השיאים בדיאלוג ברשימה. |
| AlarmManager alarmManager | תכונה שמתחברת לservice של המערכת של התזמון של ההתראות. |
| PendingIntent pendingIntent | תכונה שאחראית לשלוח את המשתמש כשהוא לוחץ על ההתראה לMain activity. |
| Game game | מופע של הclass שאחראי לתפעל את המשחק. |
| protected void onCreate(Bundle savedInstanceState) | פעולה מובנת של המערכת שמופעלת כשהאפליקציה נפתחת. |
| private void init() | פעולה שמאתחלת את האובייקטים ומעדכנת את הSharedPreferences, בנוסף היא קוראת ל startNotificationבפעם הראשונה בשביל להגדיר את ההתראה. |
| public boolean onCreateOptionsMenu(Menu menu) | פעולה שיוצרת את התפריט. |
| public boolean onOptionsItemSelected(@NonNull MenuItem item) | פעולה כשלוחצים על אפשרות בתפריט. |
| public void onClick(View v) | פעולה שמופעלת כשלוחצים על כפתור, פה היא מטפלת במקרה כשולחצים על כפתור הstart, כפתור הstop וכתור הsovled שנמצא בדיאלוג שקופץ כשפותרים את הלוח |
| ActivityResultLauncher<Intent> StartForResult | אובייקט שאחראי לעשות דברים כשחוזרים מintent, |
| public void onActivityResult(ActivityResult result) | פעולה בתוך class אנונימי שמופעל מתי שחוזרים מintent ופה מאתחלים את המשחק כשחוזרים מהintent. |
| public void startNotification() | פעולה שמפעילה חזרה של התראה לשחק כל יום ב20:20, הפעולה נקראת רק פעם אחת ויותר לא קוראים לה. |
| protected void onPause() | פעולה מובנת שמופעלת כשיוצאים מהactivity וכשהיא נקראת עוצרים את המשחק(stop mode) |
| public List<Record> createRecordListForShow() | פעולה שיוצרת רשימה של השיאים ואם יש יותר מ10 שיאים היא מצמצמת את הרשימה ל10 שיאים הראשונים. |
| public void createRecordsDialog() | פעולה שיוצרת את הדיאלוג של השיאים, ובתוכה יש פעולת onClick() שמטפלת במקרה שלוחצים על הכפתור btnOrder. |

**SettingActiviry:**

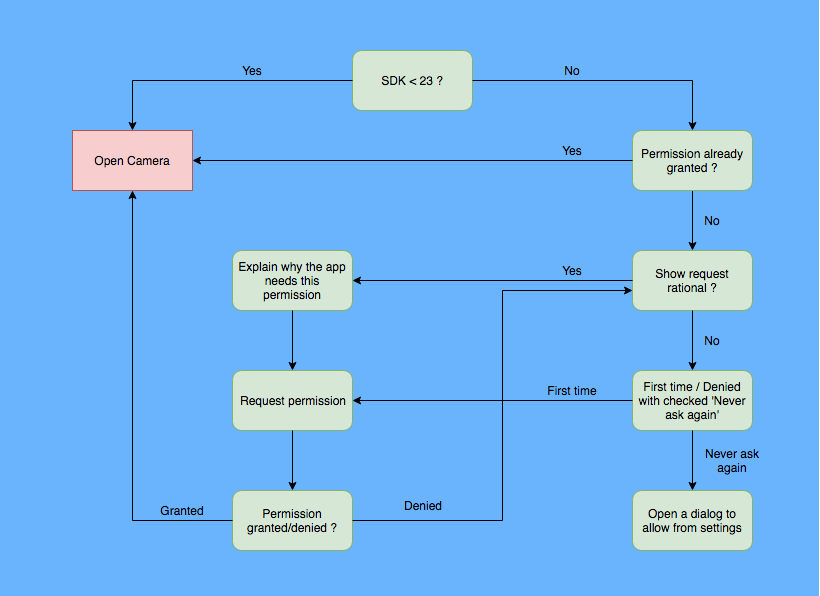
מחלקה של activity של הגדרות שמתחברת לקובץ xml – activity\_setting.xml ובה יש הגדרות לאפליקציה של שינוי גודל הלוח ושינוי הצבע של האריחים בלוח.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| Spinner spinnerSize | ספינר שאחראי להציג את כל הגדלים שאפשר לשנות אליהם. |
| Spinner spinnerColor | ספינר שאחראי להציג את כל הצבעים שאפשר לשנות אליהם. |
| List<String> sizeOption | רשימה שבתוכה יש את כל הגדלים ואותה מכניסים לתוך spinnerSize. |
| List<String> colorOption | רשימה שבתוכה יש את כל הצבעים ואותה מכניסים לתוך .spinnerColor |
| int choiceOfSize | משתנה שבו שמים את הגודל הנוכחי של הלוח ומכניסים לו את הגודל החדש אם המשתמש שינה את הגודל. |
| String choiceOfColor | משתנה שבו שמים את הצבע הנוכחי של הלוח ומכניסים לו את הצבע החדש אם המשתמש שינה את הצבע. |
| protected void onCreate(Bundle savedInstanceState) | פעולה מובנת של המערכת שמופעלת כשהאקטיביטי נפתח, קוראת ל init() |
| private void init() | פעולה שמאתחלת את האובייקטים ומקבלת מהSharedPreferences את ההגדרות הנוכחיות וקוראת ל createSpinnerOfSize() ול createSpinnerOfColor() |
| public boolean onOptionsItemSelected(@NonNull MenuItem item) | מה קורה כשלוחצים על הכפתור חזרה שנמצא למעלה בAction bar. |
| public void createSpinnerOfSize() | פעולה שיוצרת את הספינר של הגדרת הגודל של הלוח. |
| public void createSpinnerOfColor() | פעולה שיוצרת את הספינר של הגדרת הצבע של האריחים. |
| public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) | פעולה שמופעלת כשהמשתמש בוחר גודל בספינר של הגדרת הגדלים או צבע בספינר של הגדרת הצבעים. |
| public void onNothingSelected(AdapterView<?> adapterView) | פעולה שנקראת מתי שלחצת על אחד הספינרים ולא בחרת כלום, לא הכנסתי בה כלום אבל חייבים לממש אותה אז היא כתובה |

**SmsActiviry:**

מחלקה של activity של שליחת sms למפתח שמתחברת לקובץ xml – activity\_sms.xml ובה המשתמש שולח הודעת sms למפתח.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| SharedPreferences sp | תכונה שאחראית להתחבר לאחסון הפנימי ולכתוב שם נתונים |
| SharedPreferences.Editor editor | תכונה שיכולה לערוך בקובץ באחסון הפנימי |
| EditText message | אובייקט שבו כותבים את ההודעה למפתח |
| Button sendSms | כפתור שכשלוחצים עליו האפליקציה מבקשת הרשאה ואז אחרי שנותנים הרשאה הsms נשלח |
| boolean isGranted | משתנה שאומר אם ניתנה הרשאה או לא |
| protected void onCreate(Bundle savedInstanceState) | פעולה מובנת של המערכת שמופעלת כשהאקטיביטי נפתח. |
| public void init() | פעולה שמאתחלת את האובייקטים |
| public boolean onOptionsItemSelected(@NonNull MenuItem item) | מה קורה כשלוחצים על הכפתור חזרה שנמצא למעלה בAction bar. |
| public void onClick(View v) | פעולה שמופעלת כשלוחצים על הכפתור sendSms. |
| public void dialog() | פעולה שבונה דיאלוג הסבר אם המשתמש לא נתן הרשאה ומנסה עוד פעם לשלוח הודעה |
| public void sendSmsFunction() | פעולה ששולחת sms למפתח |
| public void permission() | פעולה שמטפלת בבקשת הרשאה, בפעם הראשונה היא מפעילה את הפעולה המובנת requestPermissions() ואם המשתמש דוחה את הבקשה ומנסה לשלוח שוב קוראים לפעולה dialog() להסביר למשתמש למה הוא לא יכול לשלוח את ההודעה ואם הוא דוחה עוד פעם מפנית אותו להגדרות המערכת של האפליקציה( תרשים של run time permission מצורף פה בעמוד הבא) |
| public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) | פעולה שמופעלת אחרי בקשת הרשאה והמשתמש אישר או שהוא לא אישר |



Send sms

קבצי קוד – מחלקות העצמים

**Record:**

מחלקה המייצגת שיא והתכונות שלו הם: הזמן שלקח לפתור, כמות המהלכים שהוא עשה והתאריך של אותו יום.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| private int move | תכונה שמייצגת את מספר המהלכים שלקח למשתמש לפתור. |
| private String time | תכונה שמייצגת את הזמן שלקח לפתור. |
| private String date | תכונה שמייצגת את התאריך של אותו יום שבו המשתמש פתר. |
| public Record(int move, String time, String date) | פעולה בונה. |

**RecordHelper:**

מחלקה שיורשת מ SQLiteOpenHelper ומיועדת בהתחברות לבסיס נתונים מקומי בשביל להכניס אליו את השיאים של המשחקים ובשביל לקחת אותם מהבסיס נתונים בתור רשימה.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| public static final String DATABASENAME = "records.db" | תכונה שמייצגת את השם של הבסיס נתונים. |
| public static String table\_records | תכונה שמייצגת את שם הטבלה שאליה אנחנו רוצים לגשת. |
| public static final int DATABASEVERSION = 1 | תכונה שמייצגת את הגרסה הנוכחית של הבסיס נתונים. |
| public final String COLUMN\_MOVE = "move" | תכונה שמייצגת את העמודה של המהלכים בטבלה. |
| public final String COLUMN\_TIME = "time" | תכונה שמייצגת את העמודה של הזמן שלקח לפתור בטבלה. |
| public final String COLUMN\_DATE = "date" | תכונה שמייצגת את העמודה של התאריך שבו היה השיא בטבלה. |
| SQLiteDatabase database | מופע של המחלקה SQLiteDatabase ואיתו ניגשים לבסיס נתונים. |
| private static String CREATE\_TABLE\_RECORD | תכונה מסוג מחרוזת שאליה נשרשר את הפקודה בsql ליצירת טבלה. |
| private String orderBy | תכונה שמייצגת את הסדר שבו נרצה לסדר את השיאים, לפי הזמן או לפי המהלכים. |
| String[] allColumns = { COLUMN\_MOVE, COLUMN\_TIME, COLUMN\_DATE}; | מערך של string שבו יש את כל שמות העמודות בטבלה. |
| public RecordHelper(Context context, final String table\_records,String orderBy) | פעולה בונה שבתוכה גם קוראים בsuper() לפעולה הבונה של SQLiteOpenHelper וגם יוצרים את הפקודה בsql ליצירת טבלה. |
| public void onCreate(SQLiteDatabase db) | פעולה מובנת שנקראת כשרוצים ליצור טבלה חדשה או להתחבר לטבלה קיימת בבסיס נתונים. |
| public void open() | פעולה שנותנת הרשאת כתיבה בבסיס נתונים. |
| public Record createRecord(Record r) | פעולה שמקבלת שיא ומוסיפה אותו לבסיס נתונים. |
| public ArrayList<Record> getAllRecord() | פעולה שמחזירה את כל השיאים. |

**RecordAdapter:**

מחלקה שיורשת מ RecyclerView.Adapter<RecordAdapter.RecordViewHolder> ואחראית לסדר את המערך שיאים בתוך recycle view שיציג אותו בצורה גרפית יפה.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| private final Context context | תכונה שמייצגת את המקום שבו צריך לארגן את הrecycle view. |
| private final List<Record> recordList | תכונה שמייצגת את הרשימה של השיאים. |
| public RecordAdapter(Context context, List<Record> recordList) | פעולה בונה. |
| public RecordViewHolder onCreateViewHolder(ViewGroup parent, int viewType) | פעולה שיוצרת אובייקט שיסדר את שיציג את השיאים בצורה שכתובה בxml records\_layout בתוך הrecycle view. |
| public void onBindViewHolder(RecordViewHolder holder, int position) | פעולה שנקראת כמספר השיאים שנמצאים ברשימה ומכניסה אותם אחד אחרי השני בrecycle view. |
| public int getItemCount() | פעולה שמחזירה את הגודל של הרשימת שיאים. |
| public RecordViewHolder(View itemView) | פעולה שנמצאת בתוך הclass RecordViewHolder ומאתחל כל בשביל כל view את הרכיבים שנמצאים בתוכו. |

**Square:**

מחלקה המייצגת ריבוע בלוח שעליו יכול להיות אריח, בלוח של 4 על 4 לדוגמא יש 16 כמוהו.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| BoardGame boardGame | תכונה שמייצגת את הלוח שבו נמצא הריבוע. |
| Tile tile | תוכנה שמייצגת את האריח שנמצא על הריבוע או null אם על הריבוע נמצא החור במשחק. |
| float x,y,w,h | תכונות שמייצגות את המיקום של הריבוע והגודל שלו: מיקום בx,y וגודל בw,h. |
| int number | תכונה שמייצגת את המספר של הריבוע, הריבועים ממוספרים מ1 עד גודל הלוח לפי סידורם בלוח. |
| public Square(BoardGame boardGame,float x,float y,float w,float h,Tile tile,int number) | פעולה בונה. |
| public boolean didXAndYInSquare(float xo, float yo) | פעולה שמקבלת x וy ובודקת אם הם נמצאים על הריבוע. |

**Tile:**

מחלקה המייצגת אריח בלוח, כל אריח נמצא על ריבוע(Square) אחד ויש מקום אחד שחסר אריח בשביל החור לדוגמא אם הלוח בגודל 4 על 4 אז יש 15 אריחים.

|  |  |
| --- | --- |
| כותרת הפעולה/תכונה מתוך java | תיאור הפעולה/תכונה |
| BoardGame boardGame | תכונה שמייצגת את הלוח שבו נמצא האריח. |
| float x,y,w,h | תכונות שמייצגות את המיקום של האריח והגודל שלו: מיקום בx,y וגודל בw,h. |
| int color | תכונה שמייצגת את הצבע של האריח. |
| int number | תכונה שמייצגת את המספר שעל האריח, בסידור הסופי צריך שכל אריח יהיה על ריבוע עם אותו מספר. |
| Paint p, paintTitle | תכונות שמייצגות את הכלים שאיתם נצייר את האריחים ואת המספרים שעליהם. |
| public Tile(BoardGame boardGame, float x, float y, float w, float h, int num, int color) | פעולה בונה, בתוכה יש קריאה בפעולה init(). |
| init() | פעולה שמאתחלת ומגדירה את התצורה של הpaint ושל ה paintTitle. |
| public void draw(Canvas canvas) | פעולה שמציירת את האריח עם המספר שלו עליו |
| public void slide(Square to) | פעולה שמקבלת ריבוע ומעבירה את האריח הנוכחי לריבוע הזה |