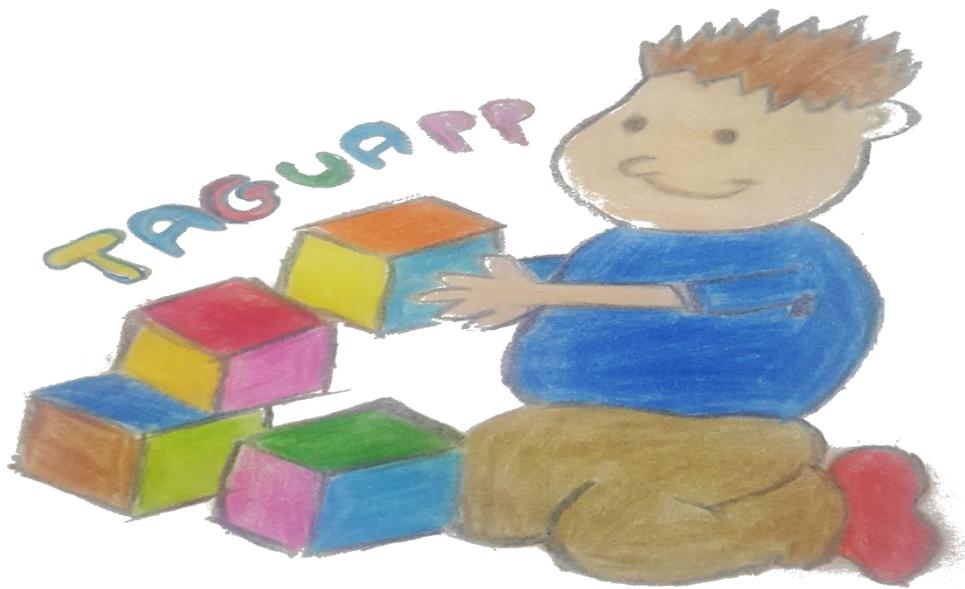


**אורט בראודה
פרויקט גמר
הנדסי תוכנה
TAGU APP
(THINK AND GROW UP)**



שם המנחה :

שאדי עסאקלה

מגייסים :

ג'MAIL נח'לה 307977090

פארס סלאמה 206163784

czherat hstodnet

ת.ז:

אני :

החותם מטה, מצהיר בזאת, שכל עבדת הגמר המוגשת בחברת זו הנה פרי עבודתי בלבד, על בסיס הנחיות של המנהה, ותור הסמכות על מקורות הידע והמידע الآخרים המצוינים בביבליוגרפיה המובאת בסוף חוברת זו.

אני מודע לאחריות שאני מקבל על עצמי ע"י חתימתן על הצהרה זו שכל הנאמר בה הינו אמת ורך אמת.

חתימת מגיש העבודה

אישור המנהה :

הרני מאשר הגשת החברת להערכתה

חתימת המנהה

אישור ראש מגמה :

הרני מאשר הגשת החברת להערכתה

חתימת ראש המגמה

czherat hstodnet

ת.ז:

אני :

החותם מטה, מצהיר בזאת, שכל עבדת הגמר המוגשת בחברת זו הנה פרי עבודתי בלבד, על בסיס הנחיות של המנהה, ותור הסמכות על מקורות הידע והמידע الآخרים המצוינים בביבליוגרפיה המובאת בסוף חוברת זו.

אני מודע לאחריות שאני מקבל על עצמי ע"י חתימתן על הצהרה זו שכל הנאמר בה הינו אמת ורך אמת.

חתימת מגיש העבודה

אישור המנהה :

הרני מאשר הגשת החברת להערכתה

חתימת המנהה

אישור ראש מגמה :

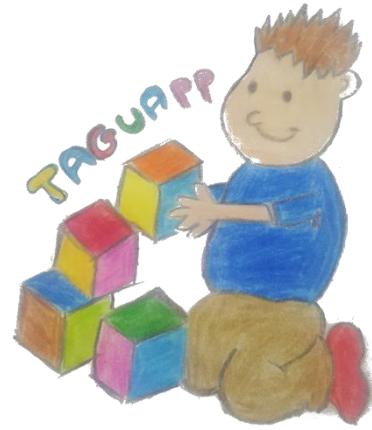
הרני מאשר הגשת החברת להערכתה

חתימת ראש המגמה

תוכן העניינים

מבוא	
5-10	
6-8	מיצוע על שפות שהשתמשו
9-10	תיאור כללי
11-50	האפליקציה
12-14	כניסה עם אינטראקציה
15-19	דף התחברות והרשמה
20-21	דף הראשי
22-25	MATH משחק
26-28	TYPING משחק
29-30	LETTERS משחק
31-32	FOCUS משחק
33-34	PHOTOS משחק
35	CUBE IT LEVEL 1 משחק
36-37	CUBE IT LEVEL 2 משחק
38	COLORS משחק
39-41	X & O משחק
42-45	CLOCK משחק
46	דף חנות - TAGUSTORE
47-48	דף INFO
49	דף ABOUT
50	דף CONTACT US
51-53	DATABASE הסבר על
54-61	פונקציות עיקריות
62-218	קוד והסבירים
63-106	קובץ XML
107-218	קובץ JAVA
219	Bibliography

מבוא



מידע על שפות שימושו בפרויקט



1. אנדרואיד :

안דרואיד זו מערכת הפעלה אשר מיועדת למכשירים ניידים ולמכשירים אשר כוח המיחשוב שלהם קטן במיוחד. אנדרואיד מבוססת על מערכת הפעלה *Linux* כERNEL שכבת קוד של ספריות ב-C-וב. C++-בשכבה מעל ניתן למצוא את ה-VM Dalvik -ה VM - אשר מסוגל להריץ קבצי Code Dalvik תכניות עבור אנדרואיד כתובים בדרך כלל ב- Java. מקור הקוד שאותו כתבנו ב Java-עובר הידור ל Code Byte Java ומיד לאחר מכן קובץ ה-Code Byte Java מועל ל VM Dalvik-ו-ניתן למשתמש למצוא את הספריות ב Java-אשר מאפשרות לנו לכתוב תכניות אשר עושים שימוש באנדרואיד.



אפליקציה (application) לאנדרואיד מורכבת מ액טיביטי (activity) אחד (או יותר). האקטיביטי (activity) אשר מהווה את ה-point-entry של האפליקציה מסומן באמצעות filter intent מותאים.

- קובץ זה מתאר את התכונות הבסיסיות של האפליקציה ומגדיר כל אחד ממרכיביה. *AndroidManifest.xml*
- ספירה זו בעבור קבצי רוד המקור. בירית המחדל היא שספריה זו תכיל את המחלקה הראשית (Activity) אשר מופעלת בכל פעם שהאפליקציה מתחילה בפעילותה.
- ספירה זו מכילה מס' ספריות משנה בעבור משאבי האפליקציה – כגון :
 - */res/drawable-hdpi* /ספריה זו מיועדת לאובייקטים ציריים (למשל : תמונה מפת סיביות). ספריות הציר אחראות מכילות משאבי המיעדים למסכים בעלי צפיפות שונה).

- `layout`/ ספירה זו מיועדת לקבצים המגדירים את ממשק המשתמש של האפליקציה.
- `values`/ ספירה זו מיועדת לקבצי XML אחרים המכילים אוסף של משתנים למשל מחוזות כתוב.

- יצירת ממשק משתמש בסיסי :
כאשר מפתחים ממשק משתמש ניתן לעשות זאת או מתוך הקוד של התכנית שאנו כתבים או באמצעות ערכתו של מסמך XML.

ניתן לכתוב קוד שככל יירה של האובייקטים שמייצגים את מרכיבי ממשק משתמש.
כך למשל : אובייקט שנוצר מהמחלקה `Button` המייצג כפתור וכך הלאה.

```
Button button = new Button(this);
```

לחילופין – ניתן לתאר את ממשק המשתמש שאנו רוצים לקבל באמצעות ערכה של מסמך XML.

כל אלמנט במסמך מייצג מרכיב אחר בownik המשמש כך למשל : XML Element XML ששמו `Button` מייצג כפתור וכך הלאה.
בזמן ריצה יוצרו אובייקטים בהתאם למסמך XML שייצרנו.

טיפול ב `events` - אירועים :
כדי לטפל ב – event שקשורים בownik המשמש – למשל לחיצה על כפתור – יש להגדיר מחלוקת שמיישמת את `OnClickListener` ולחבר אובייקט שיוצרים ממנו כך – `onClickListener` אובייקט שמייצג כפתור שאנו רוצים לטפל בחיצה עליו.

SQLite .2

באפליקציית mobile יש חשיבות רבה לשימרת נתונים, החשיבות נובעת מכמה סיבות עיקריות :

1. שימירה של נתונים עבור המשתמש, העדפות שלו, הגדרות וכו'
2. במידה ואין חיבור זמין לרשת, יש לאפשר לו לעבוד במצב לא מקוון.
3. תוכנות מחkickם `data`, של אפליקציה – יש לשמור אותם מקומיות.
4. להמעיט את הגישה לשרת לצורך אחזור של מידע קבוע, זאת על מנת לחסוך גם בסוללה וגם בעליות של מפעיל סלולרי.

ישן כמה דרכי לשמור מידע במכשיר שלנו, בפרק זה נתרכז בסיס נתונים רלוונטי מאוד נפוץ
בשם sqlite

```
class DB extends SQLiteOpenHelper {  
    final static int DB_VERSION = 1;  
    final static String DB_NAME = "mydb.db";  
    Context context;  
    public DB(Context context) {  
        super(context, DB_NAME, null, DB_VERSION);  
        // Store the context for later use  
        this.context = context;  
    }  
}
```

על מנת להתחיל לעבוד עם בסיס נתונים sqlite באפליקציית אנדרואיד אנו משתמש במחלקה שמרחיבה את SQLiteOpenHelper המגיעה עם בנאי ושתי מетодות שיש למשוך :

onCreate(), onUpgrade()

תיאור כללי

-TAGUAPP

אפליקציית משחקן חשיבה למכשירים עם מערכת הפעלה אנדרואיד המוועדת לילדים בגיל התפתחותם המאפשרת לימוד מספרים – אותיות – חישובים בסיסיים – איר השעון עובד – הקלדה – זיכרון – צבעים וכו'.

לכל משחק יש לו את המטרה והחוקים והתנאים ששhicים לו.

יש הסברים על כל משחק :

1. הגדרת המשחק – איר זה עובד.

2. מטרת המשחק.

3. איר זה עוזר ליד (מאייה בחינה זה עוזר).

4. תכנון המשחק : עיצוב + אלגוריתמיקה ופונקציות עיקריות.

נכנסנו לאפליקציה – מה עכשו ?

1. קודם כל צריך להירשם למערכת אחר כך נכנסים לדף הראשי שמציג את כל המשחקים וגם מציג פרטים של כל משתמש עם אפשרות ליצירת קשר עם המתכנתים של האפליקציה.

2. ניתן לבחור כל משחק שרצו.

3. לכל משחק יש מטרה וימוש מיוחד ופשוט לשימוש.

4. כל משתמש יכול לצפות בפרטים שלו (שם – סיסמה ומיל + אפשרות לשינוי סיסמה) בנוסף יציג את מס' הנקודות שצבר בכל משחק.

5. כל משתמש מזמין ליצור איתנו קשר במקרה של תלונה / תקלת / הצעת שיפור אפליקציה וכל דבר אחר.

6. האפליקציה מטפלת במקרה של " שכחתי את הסיסמה " - צריך לוודא שם משתמש + מיל. אם **שניהם נכונים** – אז המערכת תיתן למשתמש להיכנס לדף הראשי והוא בוחרת באופן אקראי סיסמה חדשה וכמוון ניתן לשנות אותה בדף ה - INFO

7. בדף הראשי – ניתן לעצור את השיר שברקע ע"י לחיצה על כפתור הקול בនוסף לזה ניתן גם לבחור מרשימה של השירים שיש ע"י לחיצה ארוכה על כפתור הקול.

8. בדף הראשי – כשלוחצים על כפתור "יציאה" חוזרים לדף כניסה/הרשמה.

9. כל משתמש יכול להיכנס לחנות המשחקים ולקנות את הנטבות של כל משחק.

כל משחק יש לו הטריות שונות ממשחק אחר שעוזרות למשתמש לשחק ולצבור מקסימום נקודות במינימום עבודה וזמן ומצמצמות מקרים שבהם המשתמש מפספס נקודות במהלך המשחק.

10. בדף About ישנה הסבר על האפליקציה ומ' הם המפתחים.

11. הטריות :

משחק מתמטיקה : הטרת הבבוב + הטרת הצגת תשובה.

משחק תМОנות : הטרת זמן + הטרת תМОנות חדשות

משחק שעון : הטרת עזר לקו השעון

משחק צבעים : הטרת לדג + הטרת להגיד שוב פעם

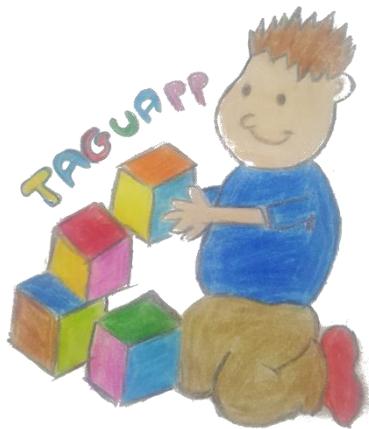
משחק הקלה : הטרת הקדלות חצי מילה + הטרת הקטנת מהירות.

.12

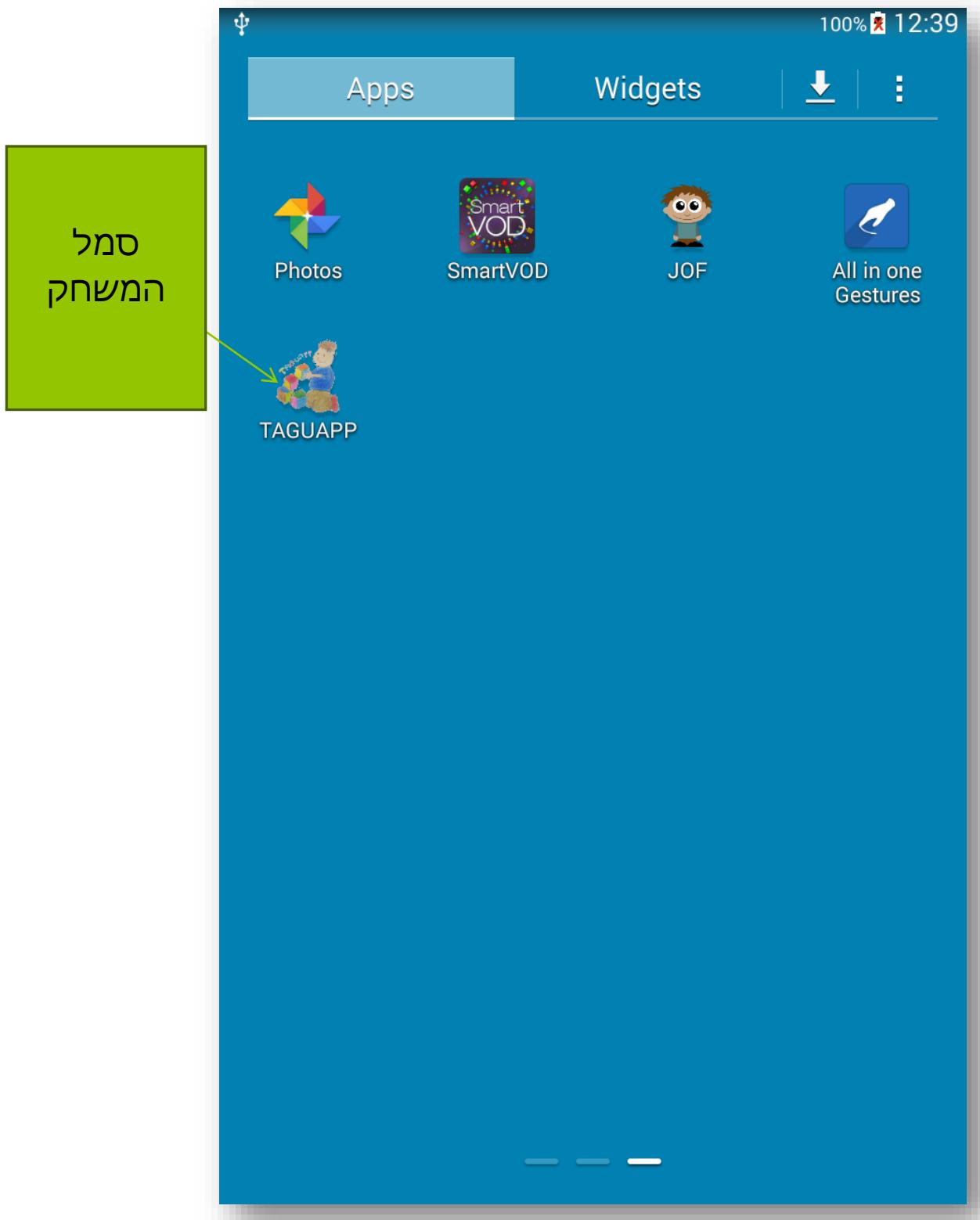
יש 3 קזבים MP3 :

- song.mp3 // מוגדר כברירת מחדל בהתחלה
- song2.mp3 // שיר מס' 2
- song3.mp3 // שיר מס' 3
- wrong.mp3 // קטע של "שגיאה" כאשר טועים במהלך המשחק השעון

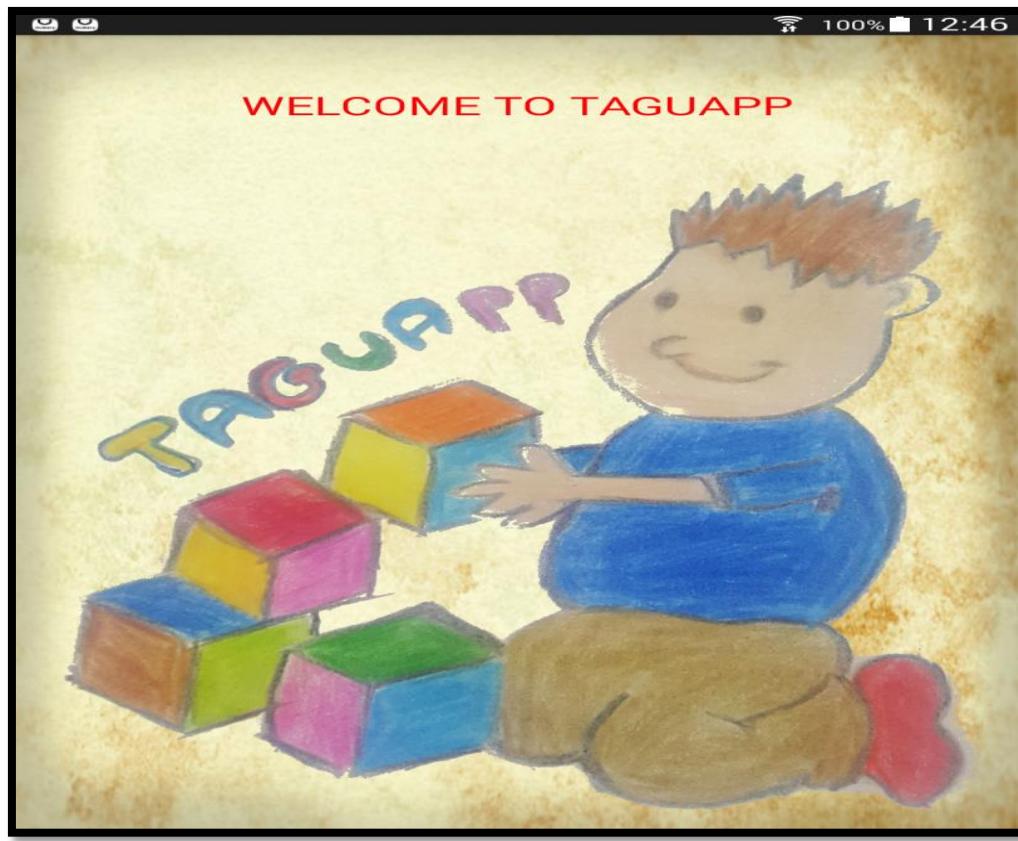
האפליקציה

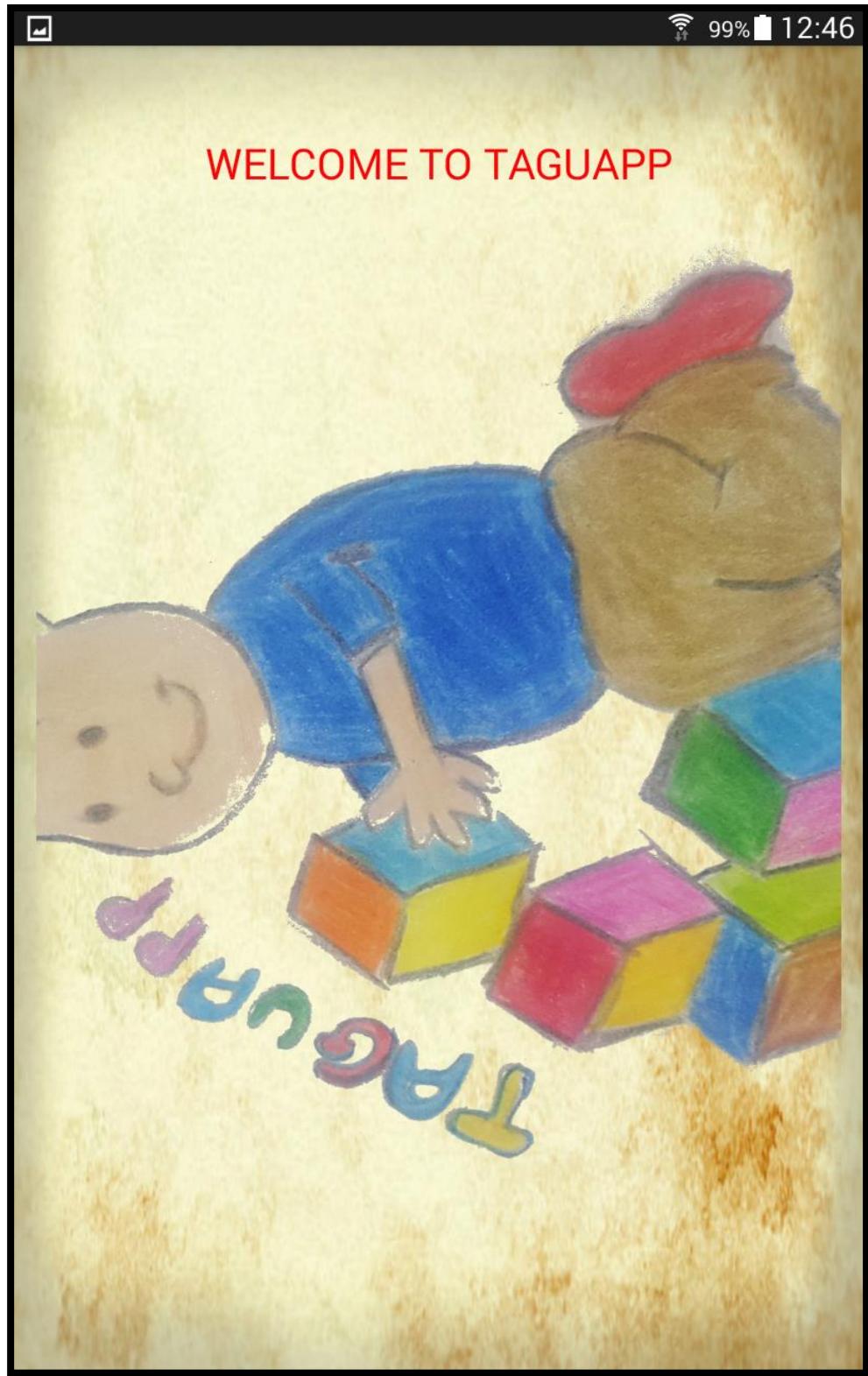


בהתחלת כארר לוחצים על משחק יתחל *Animation* לפני הכניסה.



Animation before we start



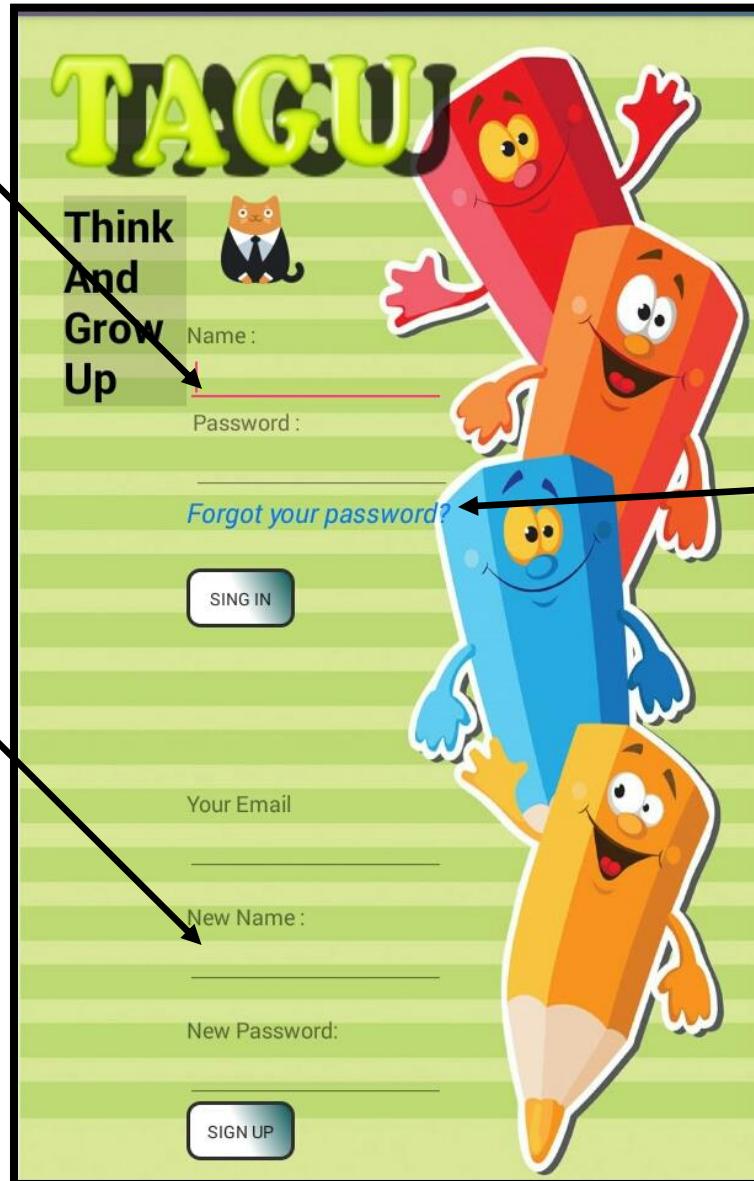


דף התחברות והרשמה

כניסה
למשתמש
קיים.

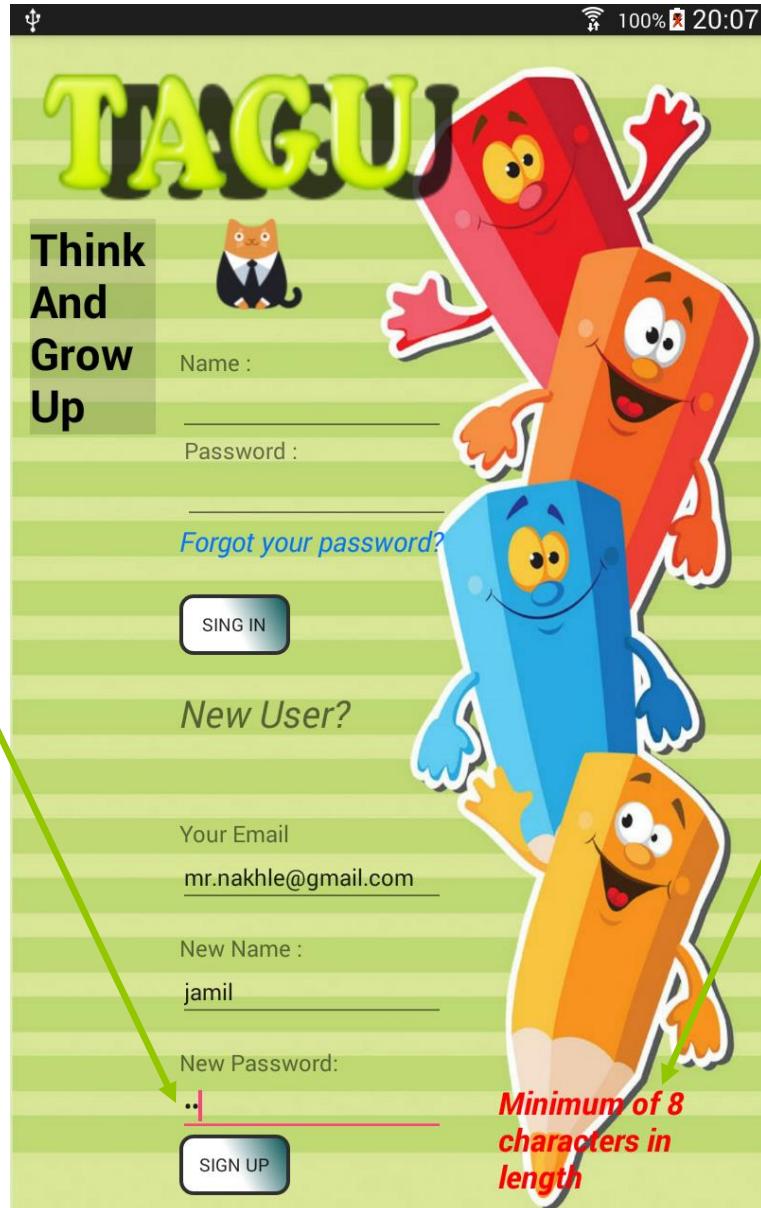
במקרה אם
שכחתי
סיסמה.

יצירת
משתמש
חדש (כולל
בדיקות
קלט)



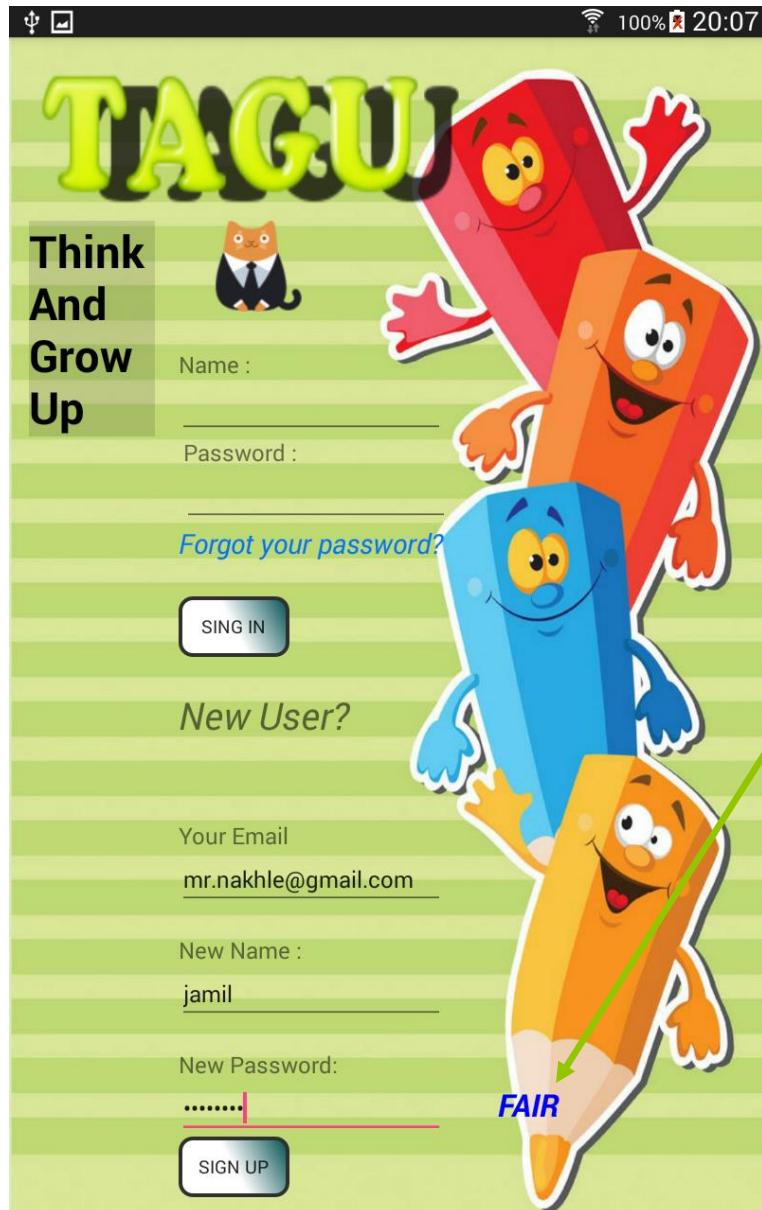
המשר SIGN IN

בדיקה
סיסמה



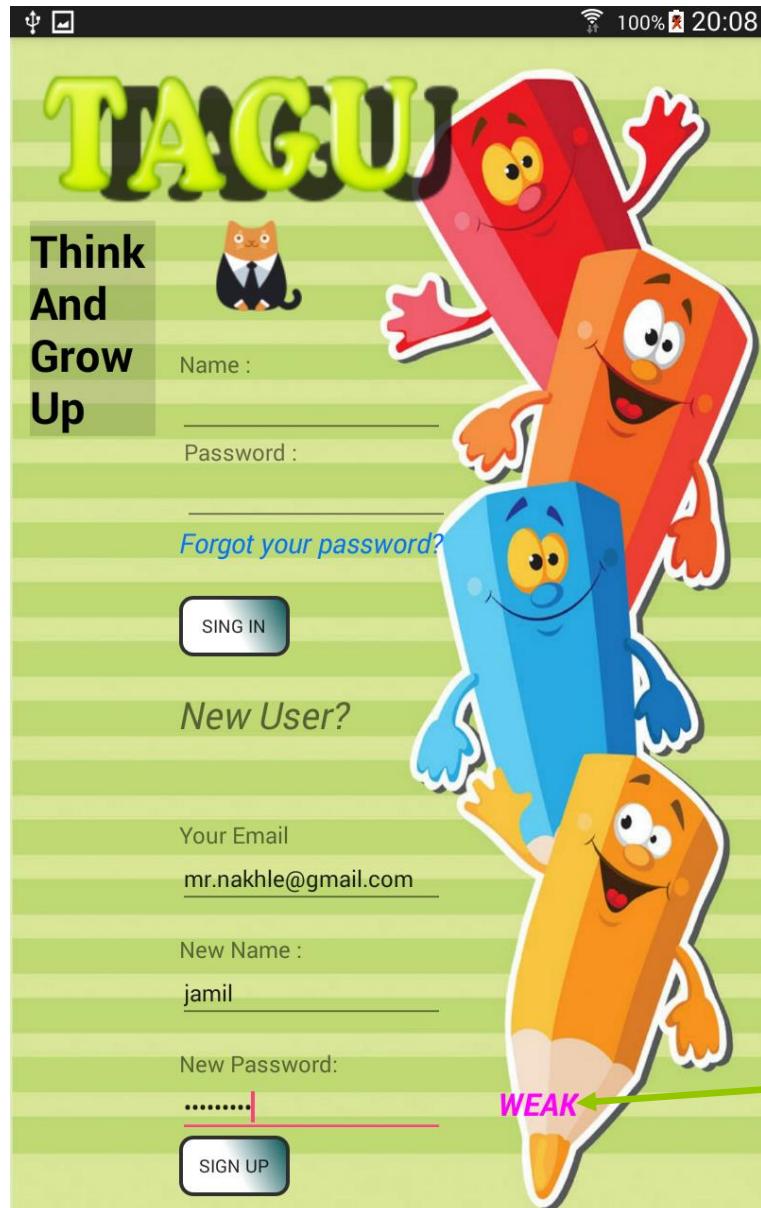
הסיסמה
צריכה
להכיל 8
תווים
מינימום

המשר SIGN IN



סיסמה
МОГДРТА С-
AM FAIR
היא באורך
8 תווים

המשר SIGN IN



סיסמה
מוגדרת כ-
אם WEAK
אורכה
על 8
תווים וגם
מכילה
אותיות בלי
מספרים.

המשר SIGN IN



סיסמה
מוגדרת כ-
STRONG
אם אורכה
מעל 8
תווים וגם
מכילה
אותיות וגם
מכילה
מספרים.

לחצן להשתקה
והפעלת שיר.
לחיצה ארוכה-
לבחירה שיר.

יש לנו
תשעה
משחקים

:

MATH

TYPING

LETTERS

FOCUS

PHOTOS

CLOCK

CUBE IT

level 1&2

COLORS

X O

דף בית APP TAGU



חנות
הטבות
TAGU STORE

פרטים של
המשתמש ב
INFO

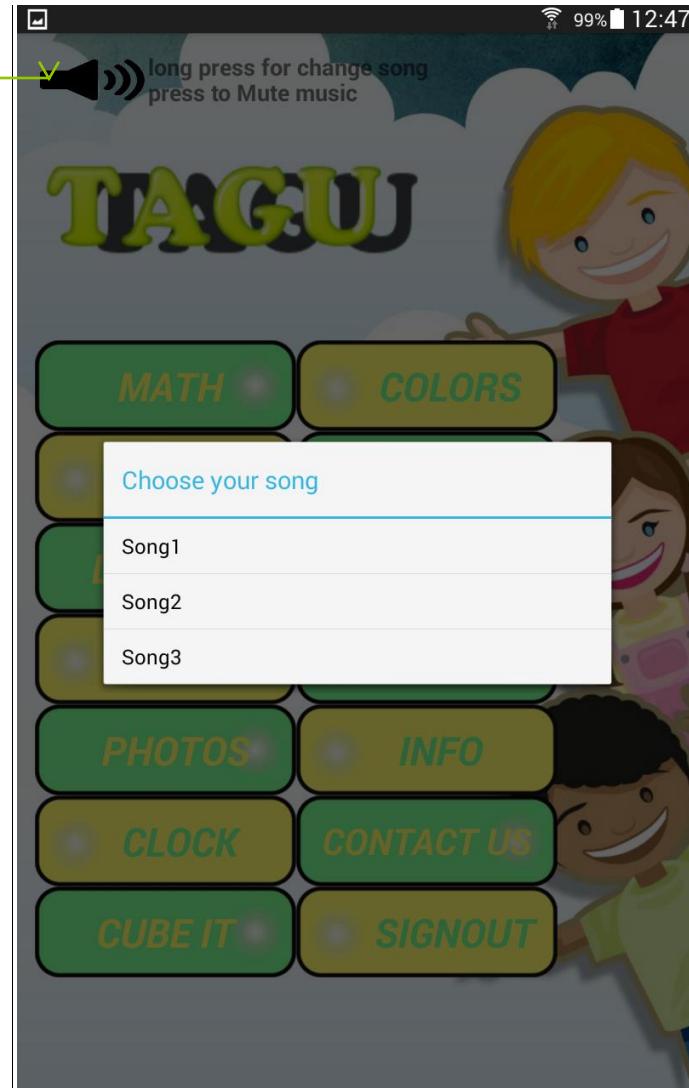
אפשרות
ליצירת איתנו
קשר דר
CONTACT US

יציאת
משתמש
SIGN OUT

דף בחירת שיר

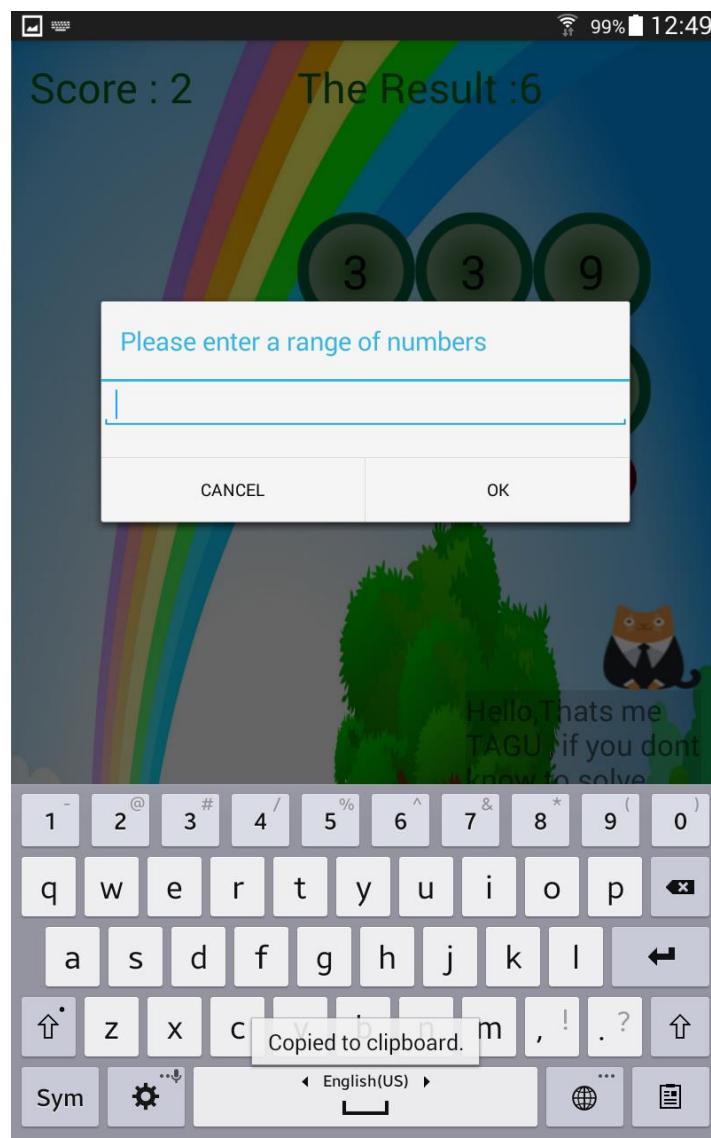
(לאחר לחיצה ארוכה על לחצן הקול)

לחיצה
לבחירה
שיר



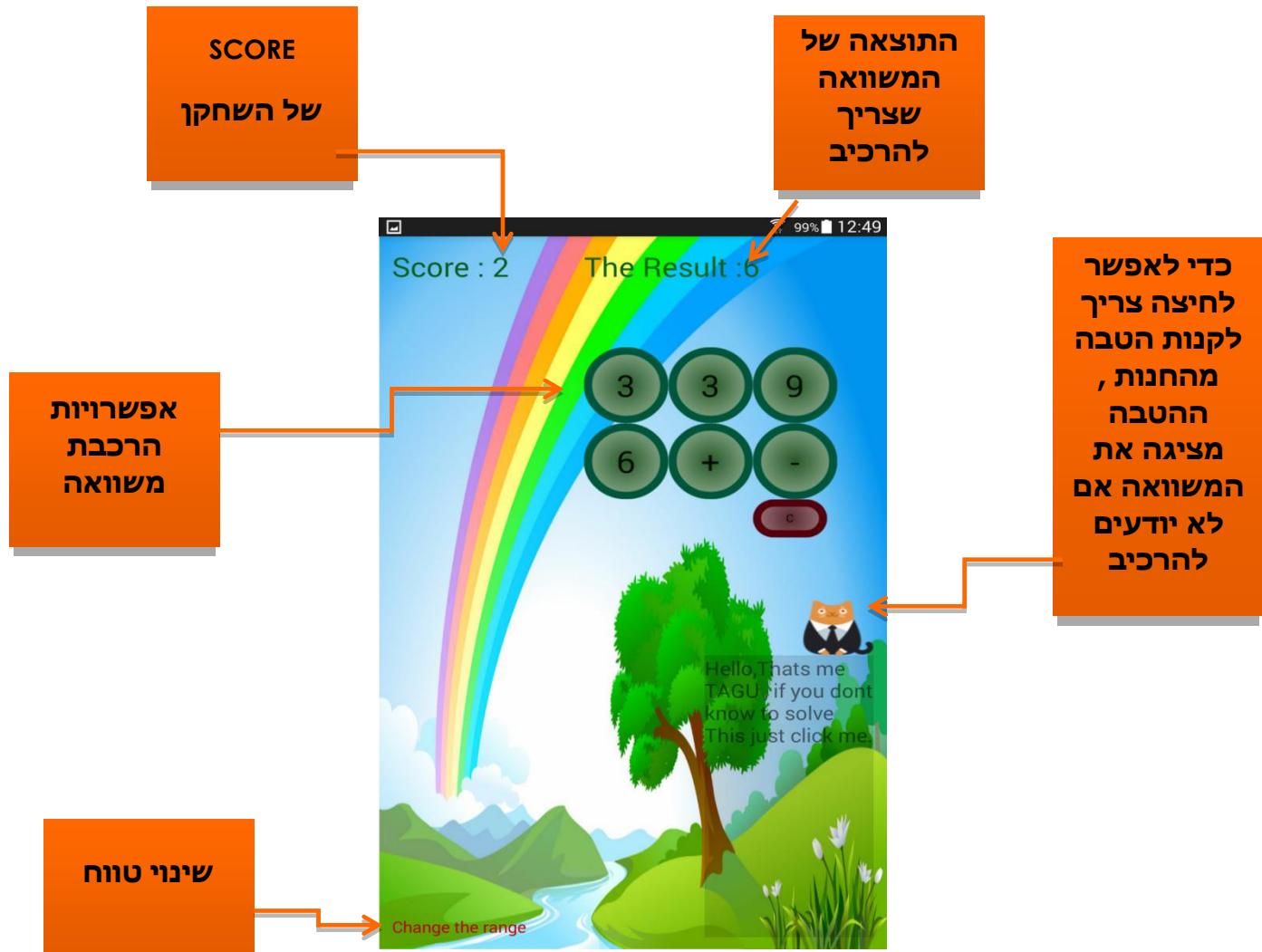
MATH

בכניסה למשחק , האפליקציה מבקשת להכניס מספר שהוא הטווח מ 1 עד המספר שנקלט, בשלב זה השחקן צריך להרכיב משווה שההתוצאה שלה תהיה כMOVEN בטוויה שנבחר.

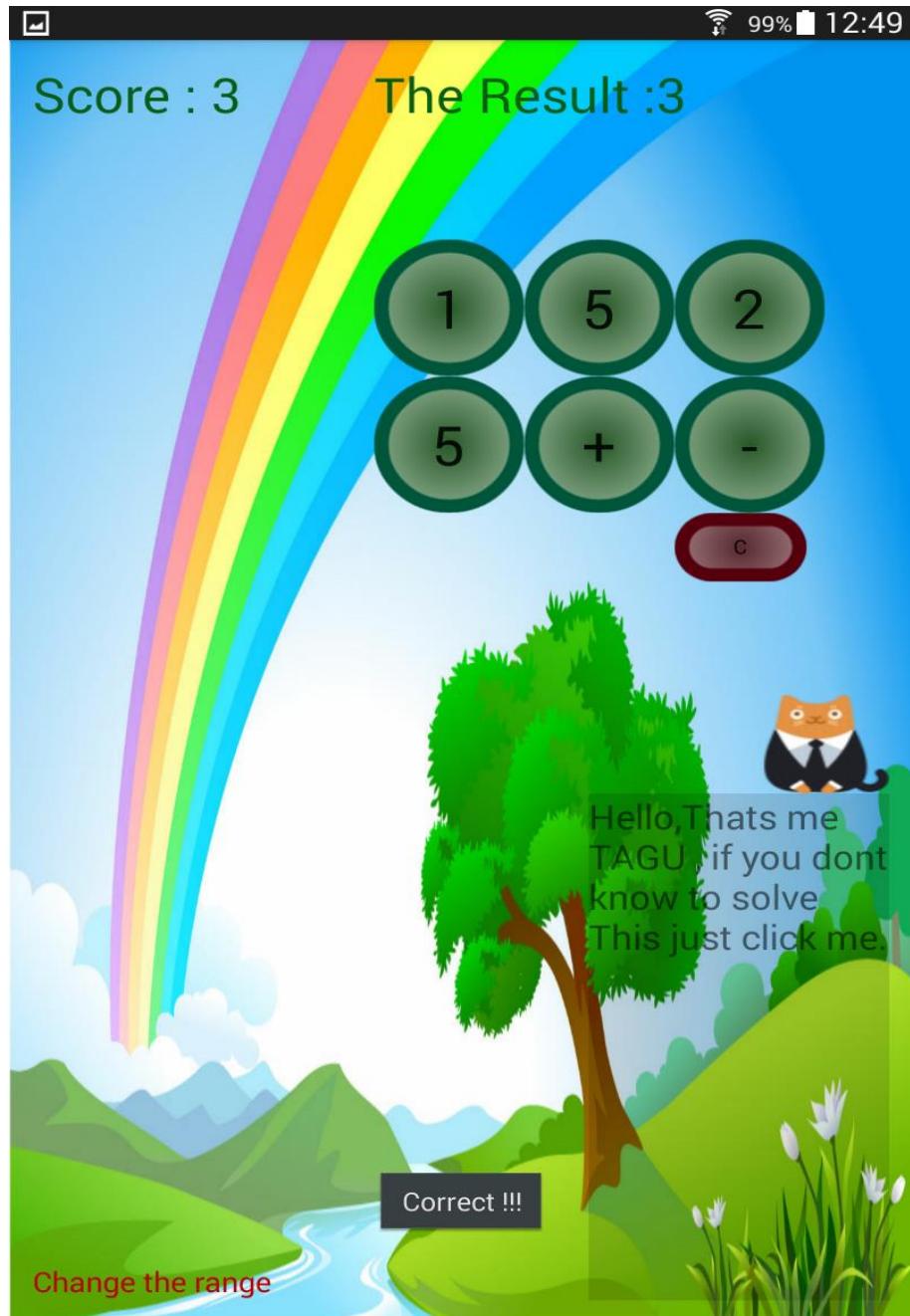


הוספנו את הטווח כדי לוודא
שכל משתמש יוכל לשחק
ולבחור את הטווח שהוא רוצה.

המשר (MATH)

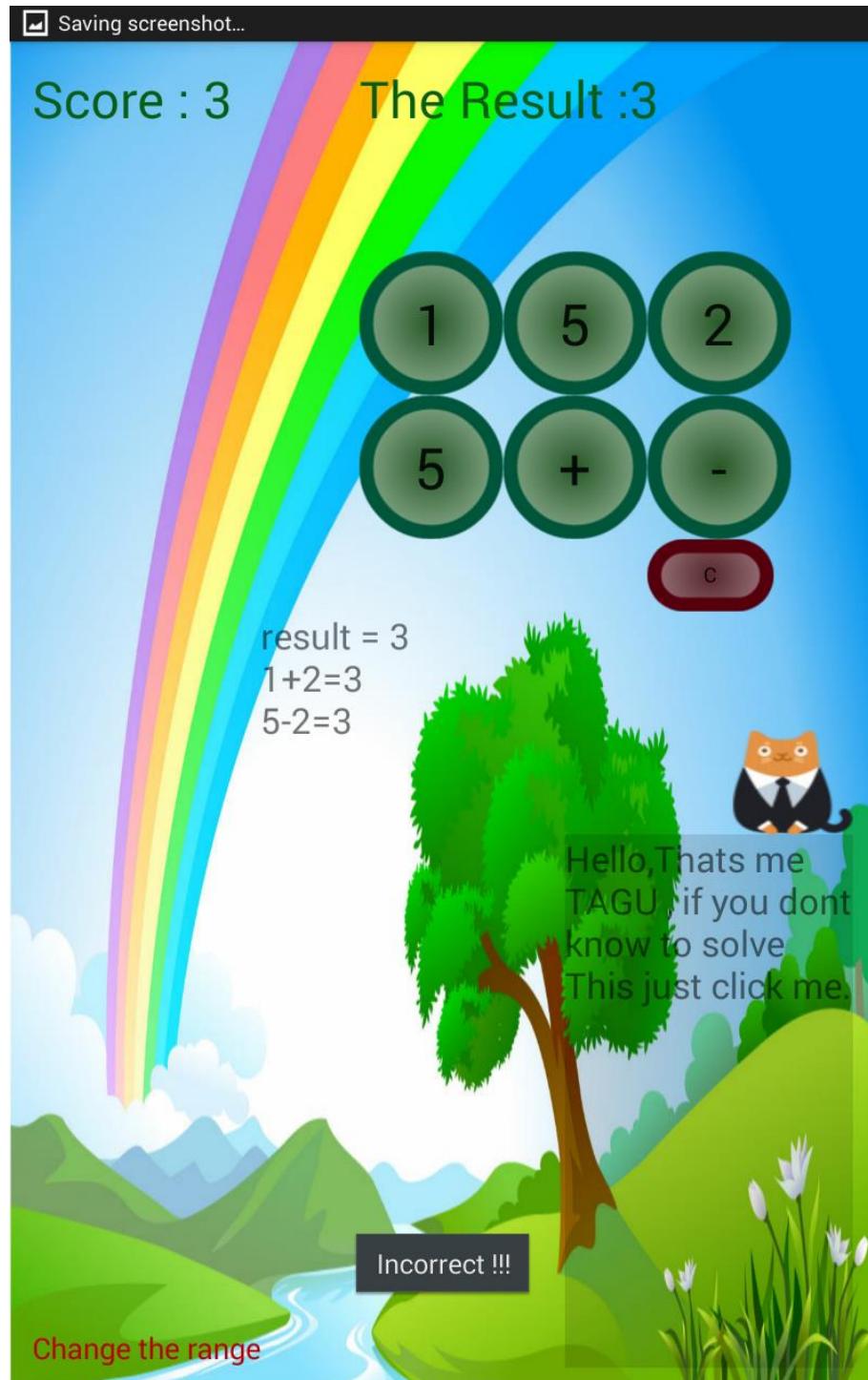


המישר MATH



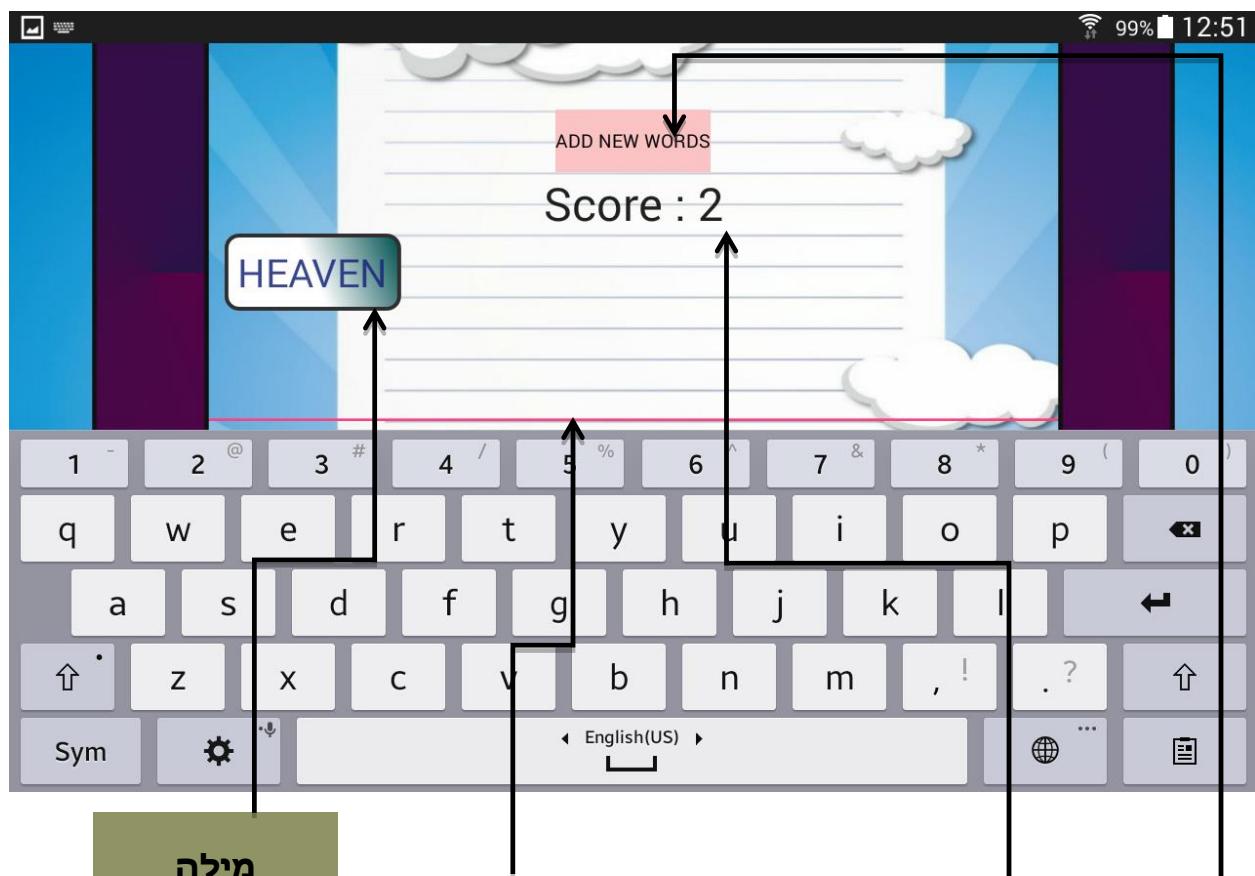
מציג הודעה
"נכון" אחרי
שהאפליקציה
בודקת את
הנוסחה שלך
אם היא
נכונה
מאפסת הכל
ומציג אותה
ההודעה
ועוברת
לשלב הבא

המשר MATH



TYPING

יש לנו מילה שעוברת מצד לצד השני, צריך להספיק לכתוב אותה לפני שתגיע לצד השני ונעלמת. זה יעזור לשחקן לשמור את המילים יותר וגם איך רושמים אותן.



מילה
שעוברת
משמאלי
לימין, צריך
לשנות את
המילה לפני
שתגיע לצד
ימין

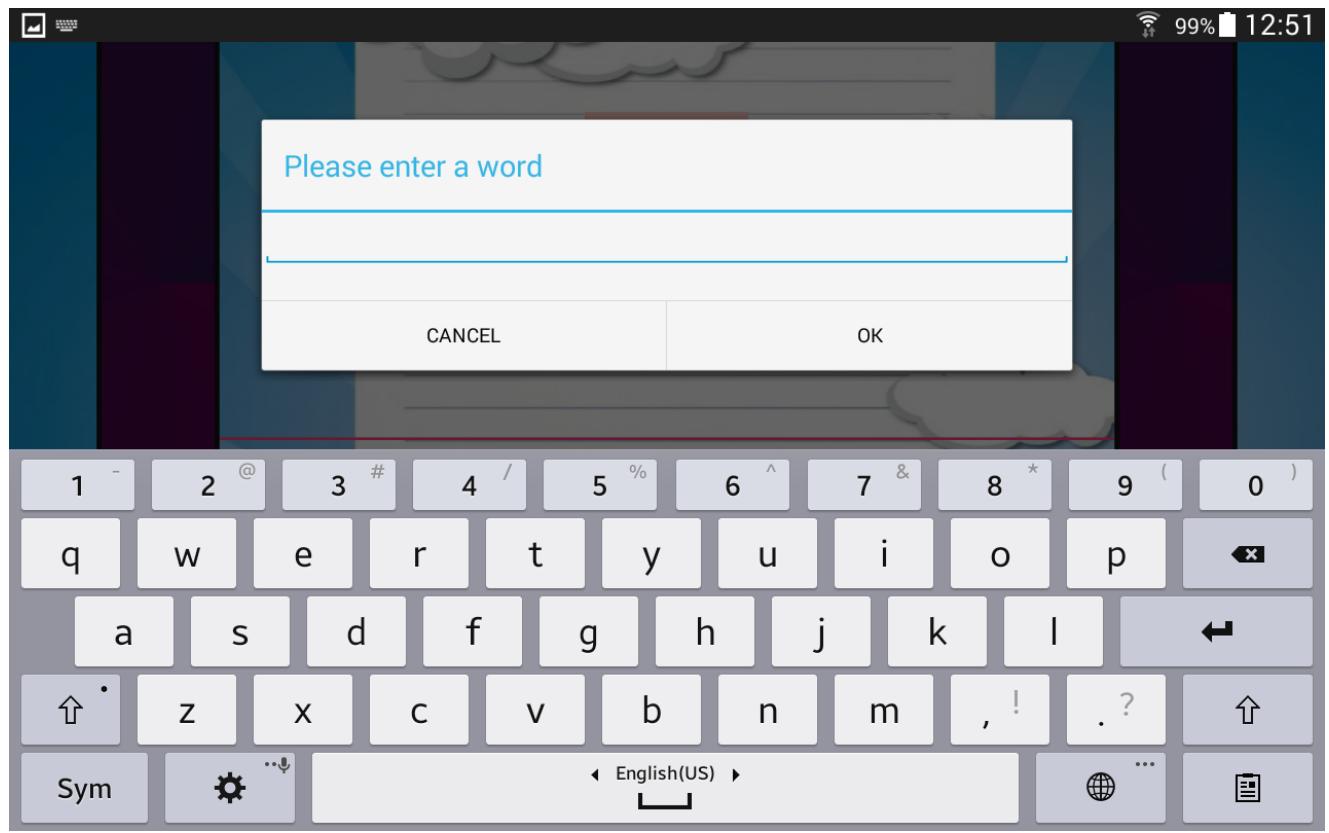
מקום
להקדלה-
הימלה-
לחיצה
ארוכה
לשינוי
FONT

SCORE
של משתמש

קליטת
מילים
חדשנות

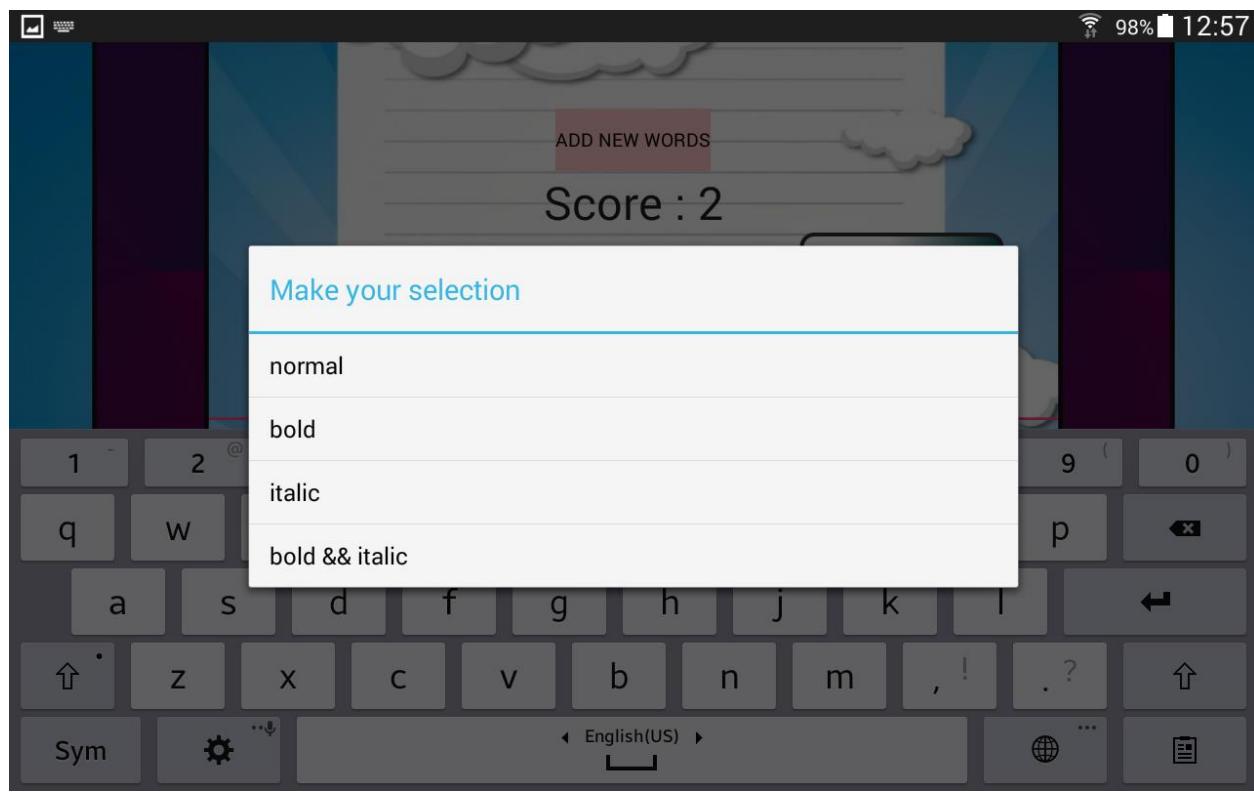
המשר TYPING

לחיצה על כפתור
מציג לנו חלון להכנסת מילים חדשות
ונשמרות לכל משתמש המילים שלו.



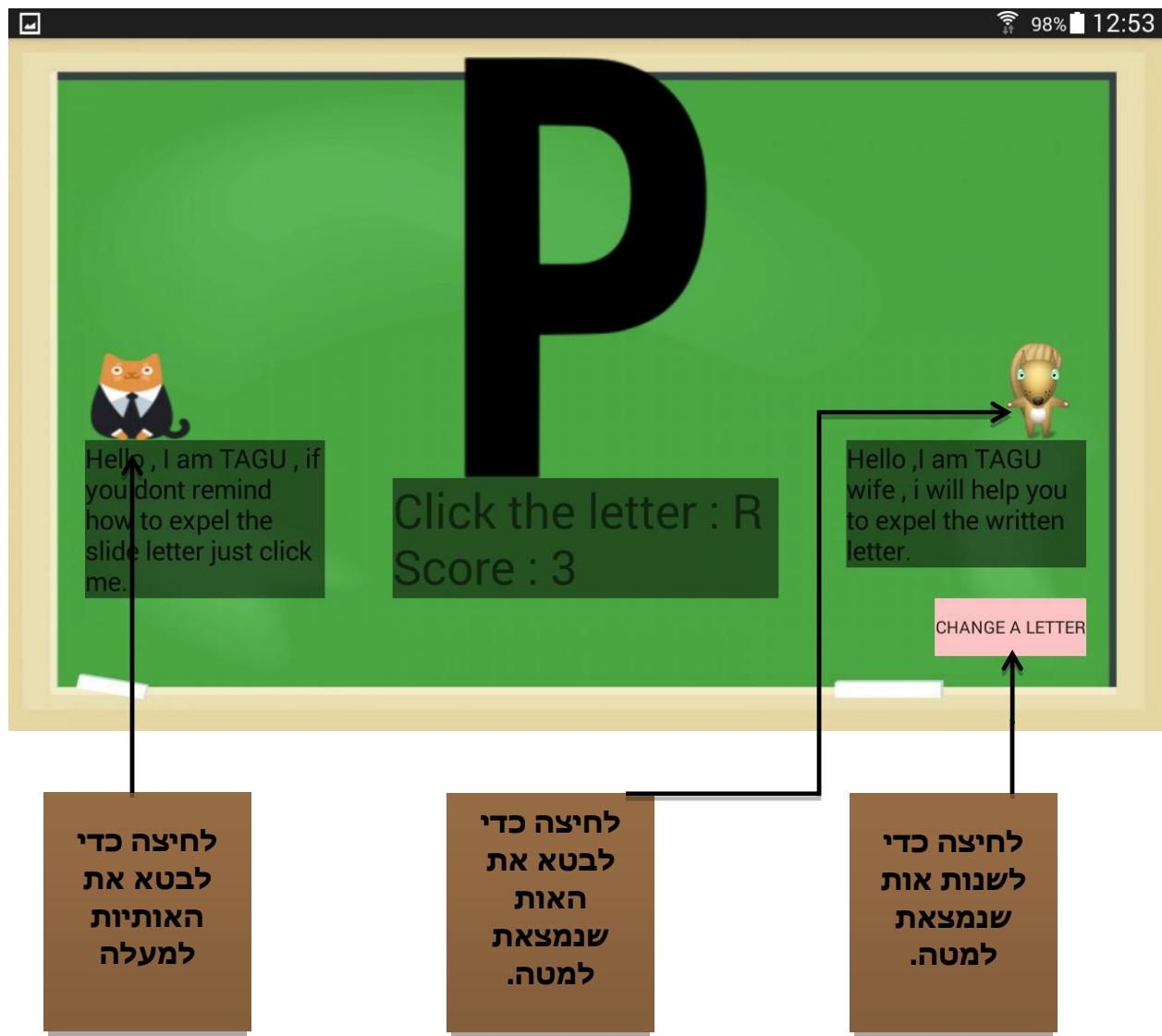
המשור TYPING

לחיצה ארוכה על תיבת הטקסט יוצג חלון לבחירת סוג כתוב

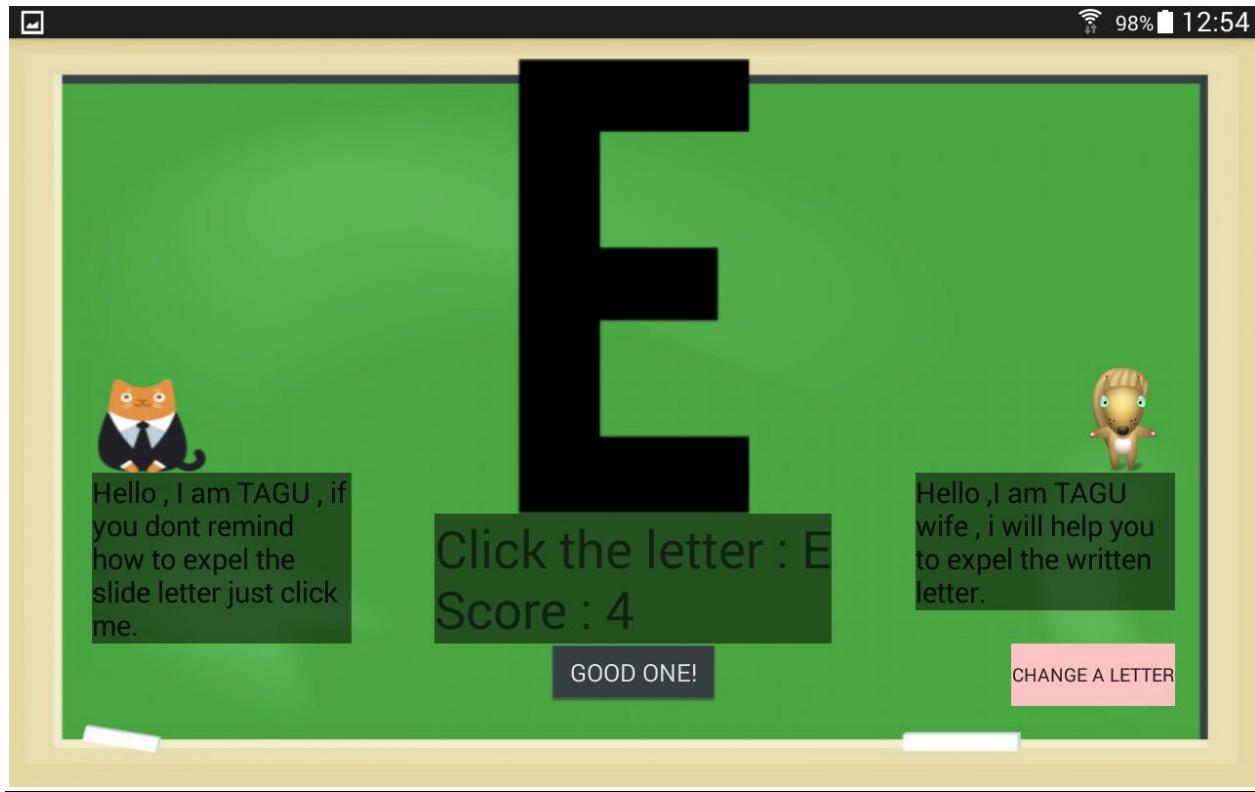


LETTERS

משחק זה נועד כדי ללמד את האותיות ואיך
לבטא אותן, למטה יש לנו אות ומעליה רצף של
אותיות שכל אחת מהן מופיעה לשניה ונעלמת,
צריך לחכות שברצף למעלה תופיע אותה אות
שנמצאת למטה וללחוץ עליה.

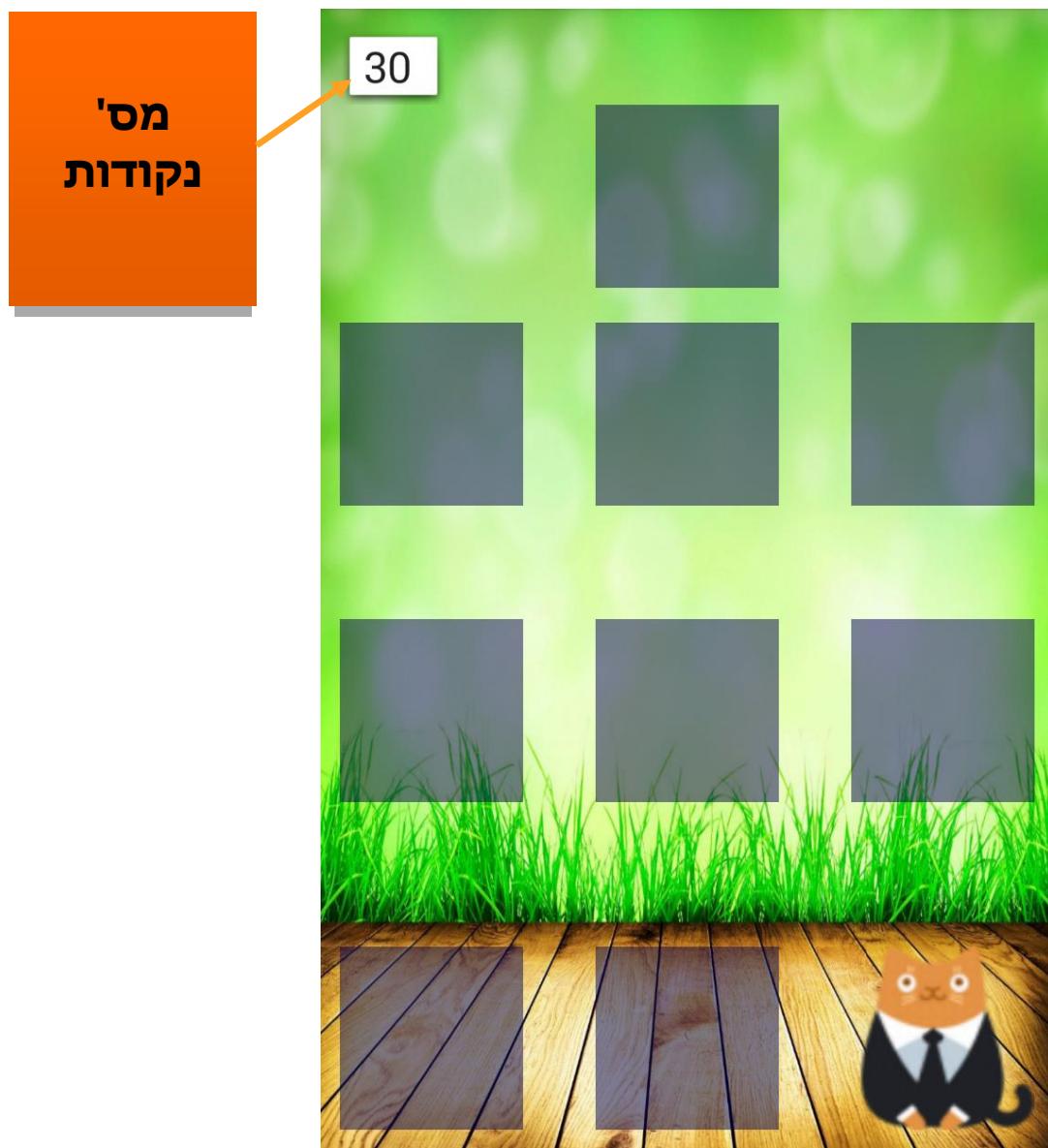


LETTERS המשור

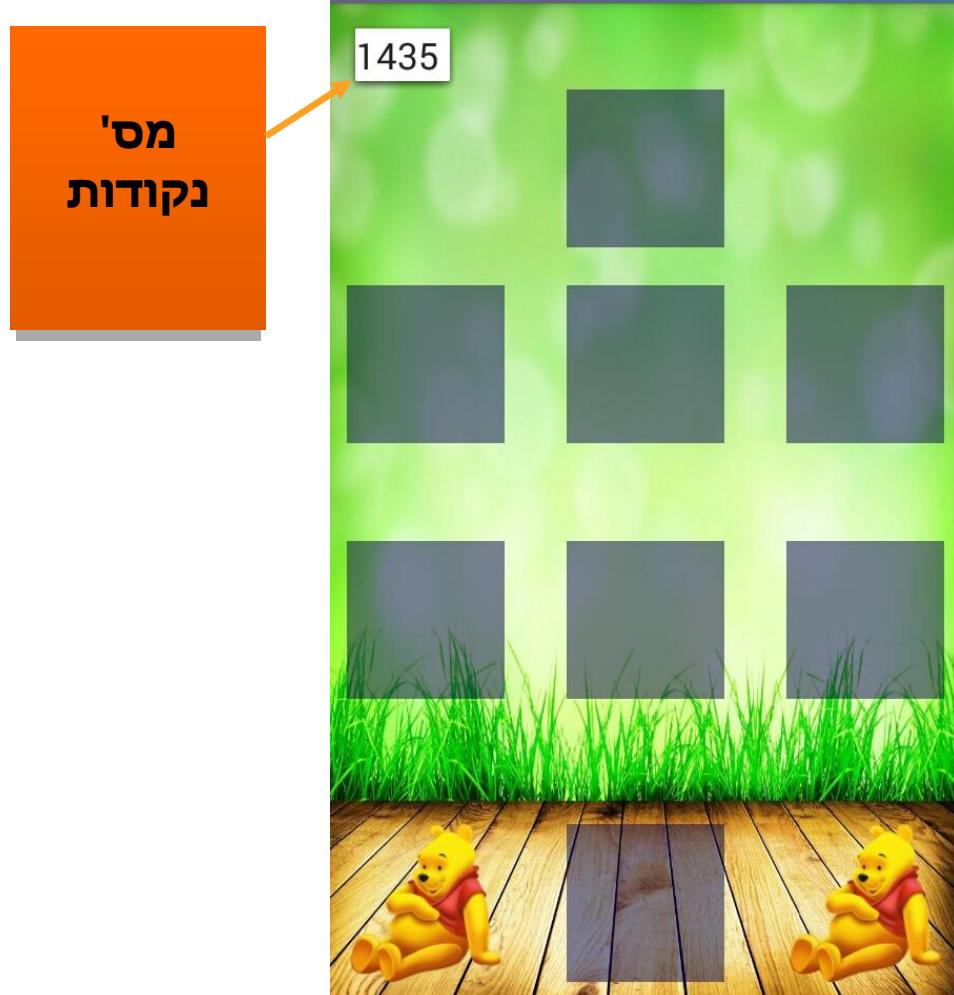


FOCUS

משחק זה נועד כדי לשפר את מהירות תגובתו של השחקן וכל מה שצריך לעשות הוא ללחוץ על תמונה לפני שהיא נעלמת, מהירות של הימלמות התמונה וכמות תמונות משתנה לפי נקודות ושלב המשחק.



המישר FOCUS



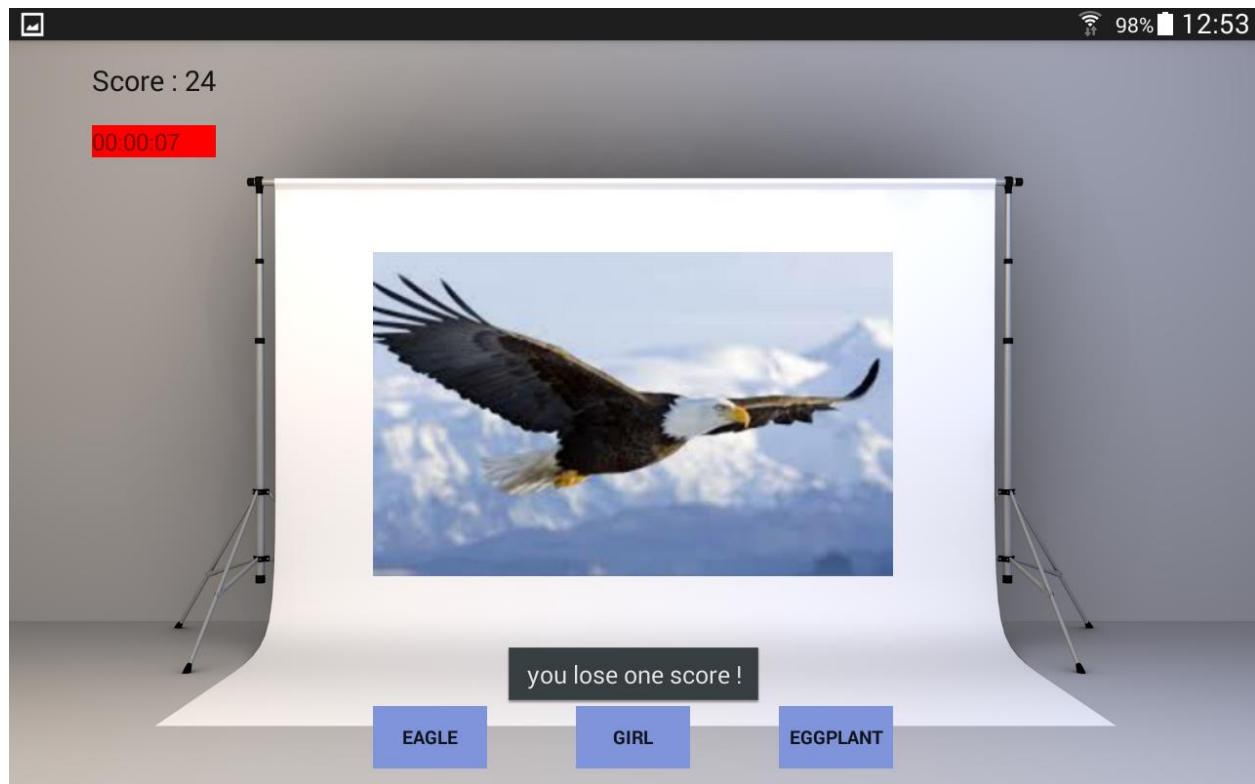
PHOTOS

**שלוש אפשרויות ותמונה אחת ותשובה
נכונה אחת - צריך לבחור תשובה
מהאפשרות לפני שזמן יגמר, שלב זה
נועד כדי לחשוף את השחקן על כל מיני
אובייקטים שונים ומשפר את עובdotו עם
הגבלה בזמן.**



הmarsh PHOTOS

**במידה והזמן נגמר ולא בחרנו אף תשובה
תוצע לנו הودעה שהפסדנו נקודה אחת
ועוברים לשלב הבא**



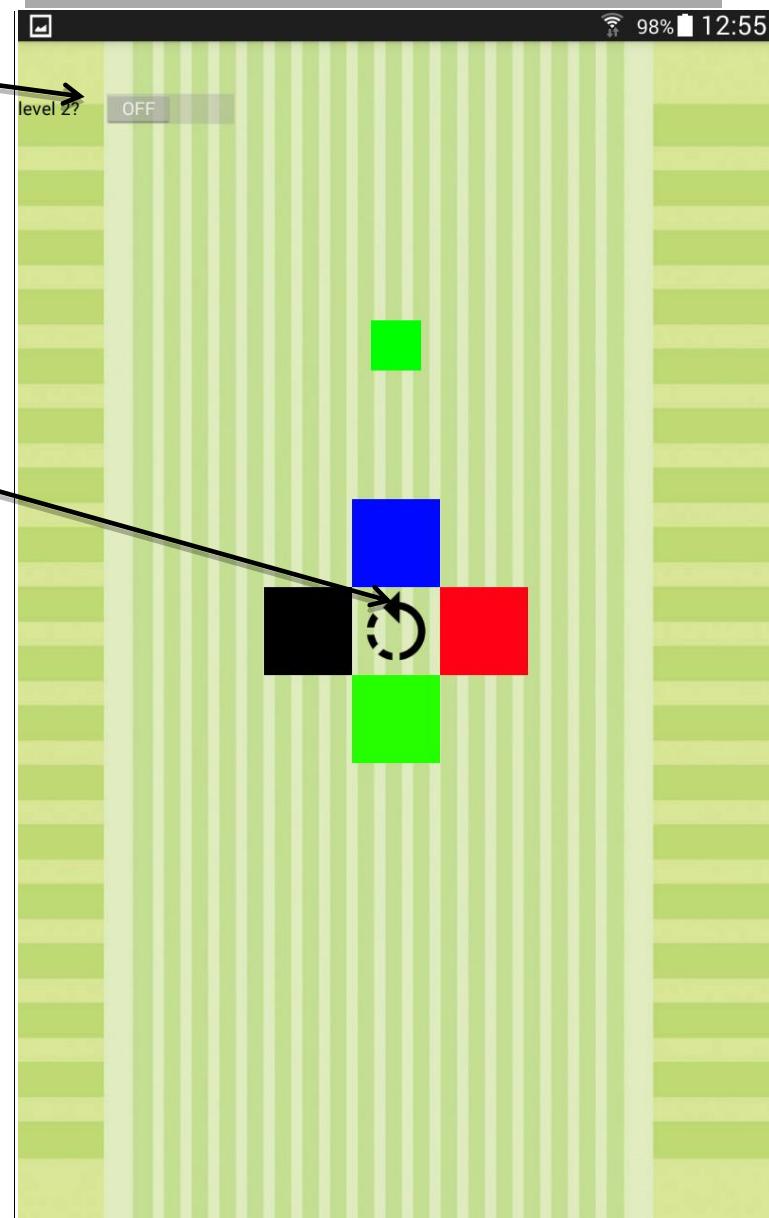
CUBE IT

משחק זה מורכב משני שלבים, יש
קוביות שmagiuot מלמעלה
ומלמטה, צריך להתאים את
הצבע לפניו ההתנגשות.

מעבר
לשלב 2
LEVEL 2

לחוץ
סיבוב כדי
לשנות
המקומות
של
הצבעים

שלב 1
LEVEL 1



המישר

(CUBE TWO)

LEVEL 2

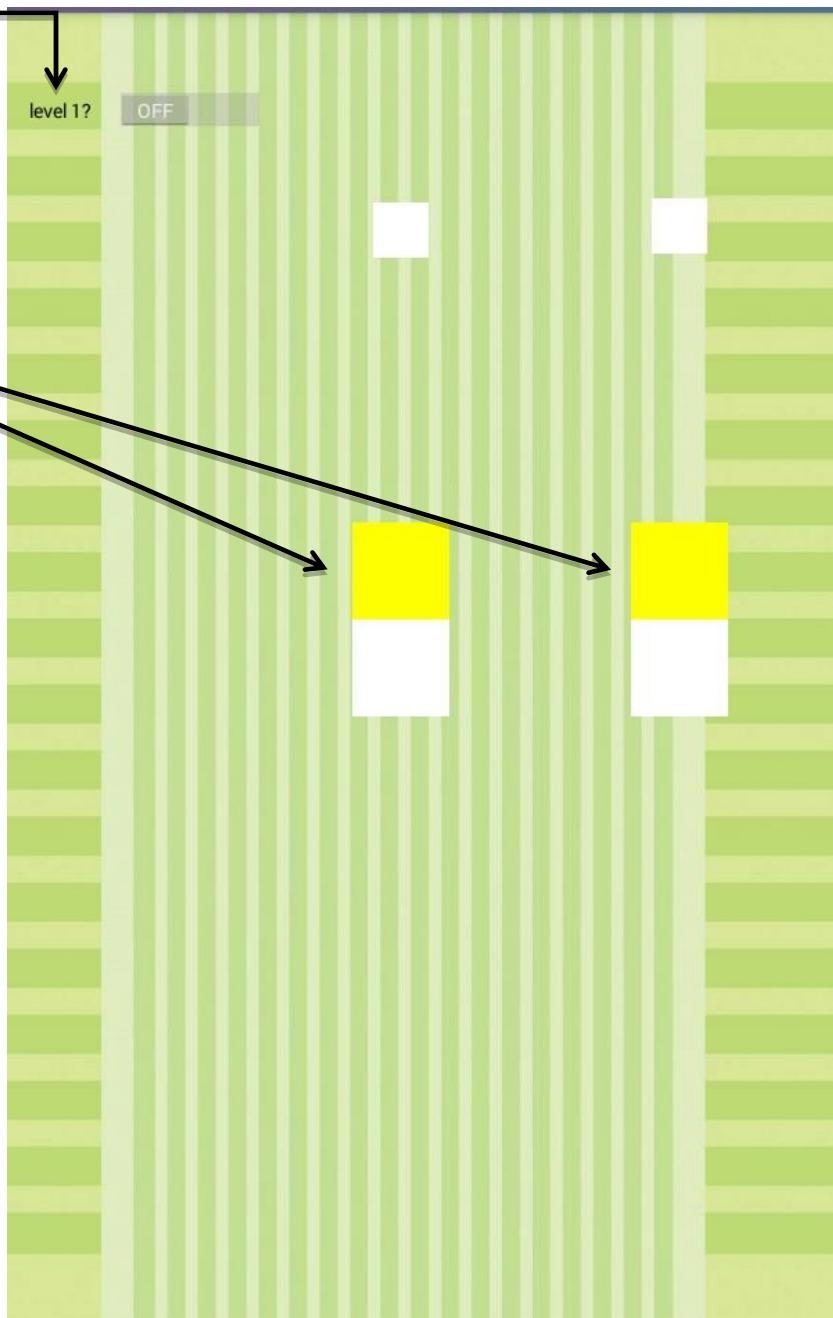
חירה
לשלב 1

level 1?

OFF

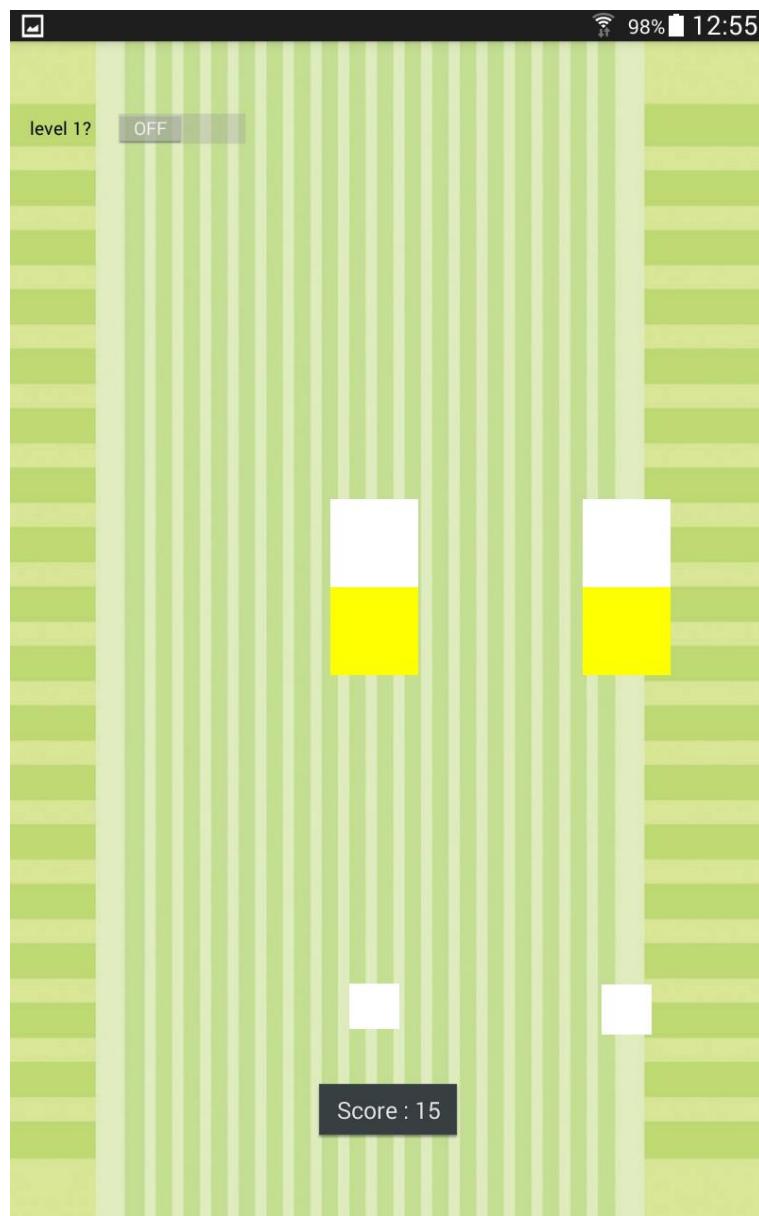
לחיצה
על
הצבעים
הופכת
מיקום
הצבעים

שלב 2
LEVEL 2



CUBE TWO **המשור**

**במידה ומרוחחים נק' תוצג לנו הودעה על הסכום
וממשיכים במשחק**

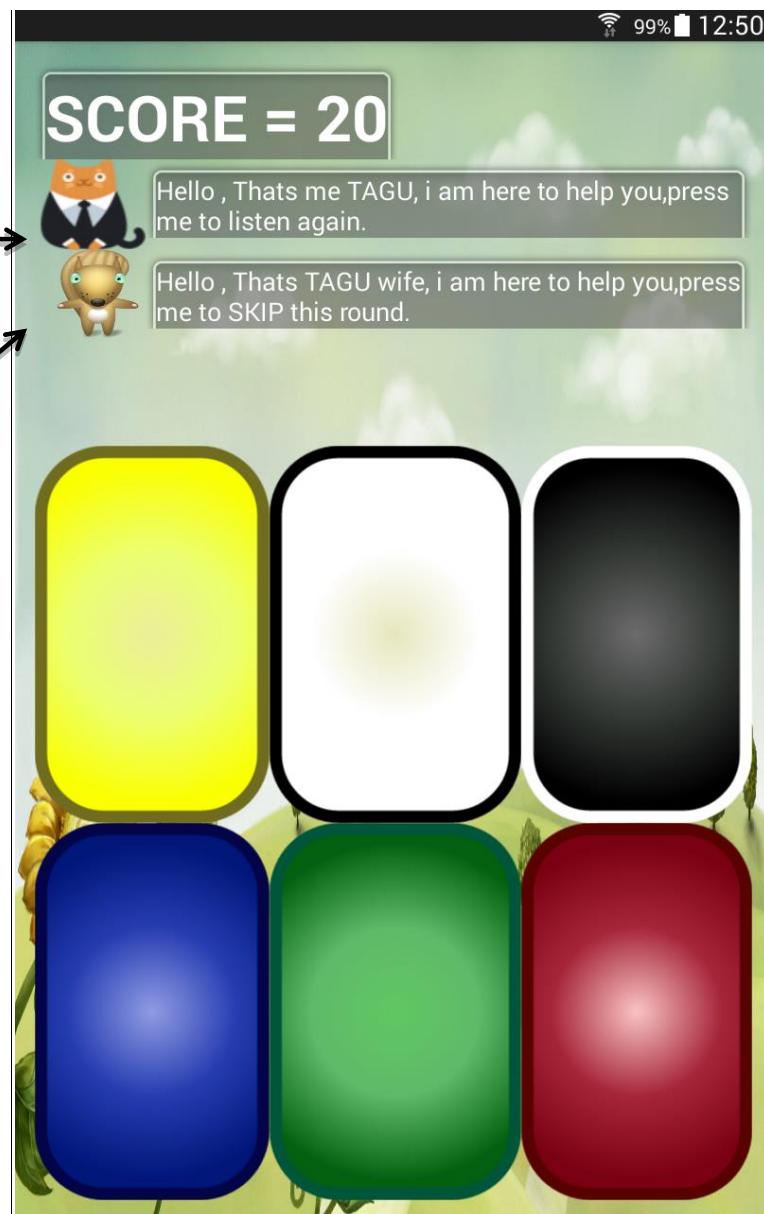


COLORS

במשחק זה נדרש להקשיב ולזכור סדר של צבעים, לאחר ההקשבה נדרש ללחוץ על צבעים בהתאם לסדר שהמענו

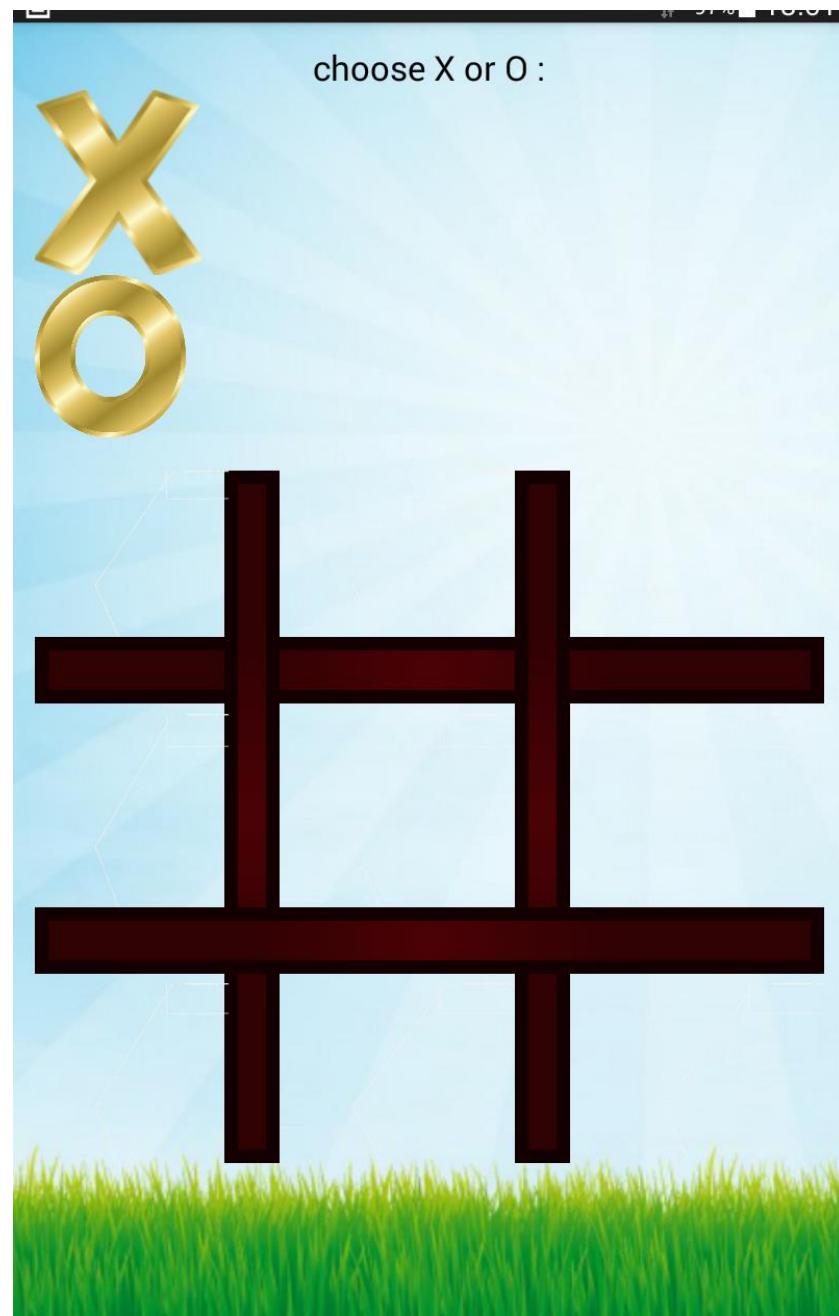
שמעה חזרת,
צריך לזכור
הטבה

Skip Feature
-
רץ חדש -
צריך לזכור
הטבה נוספת



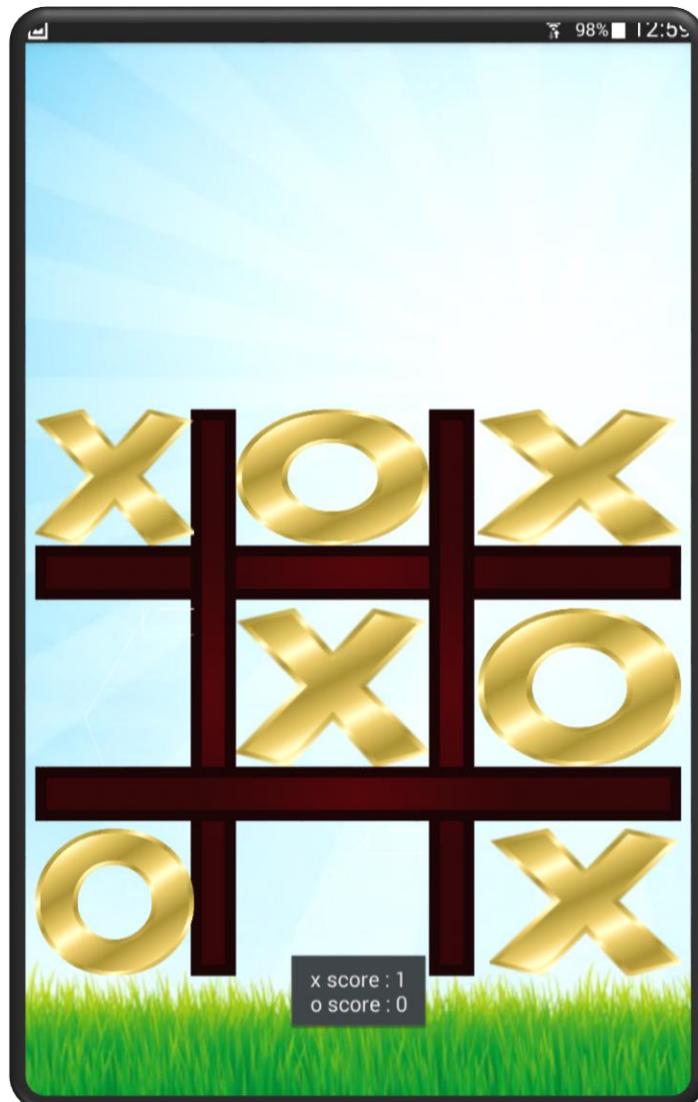
X O

**בחירה איקס או עיגול , צד שני משחק
בצורה אוטומטית.
אפליקציה X VS משתמש**

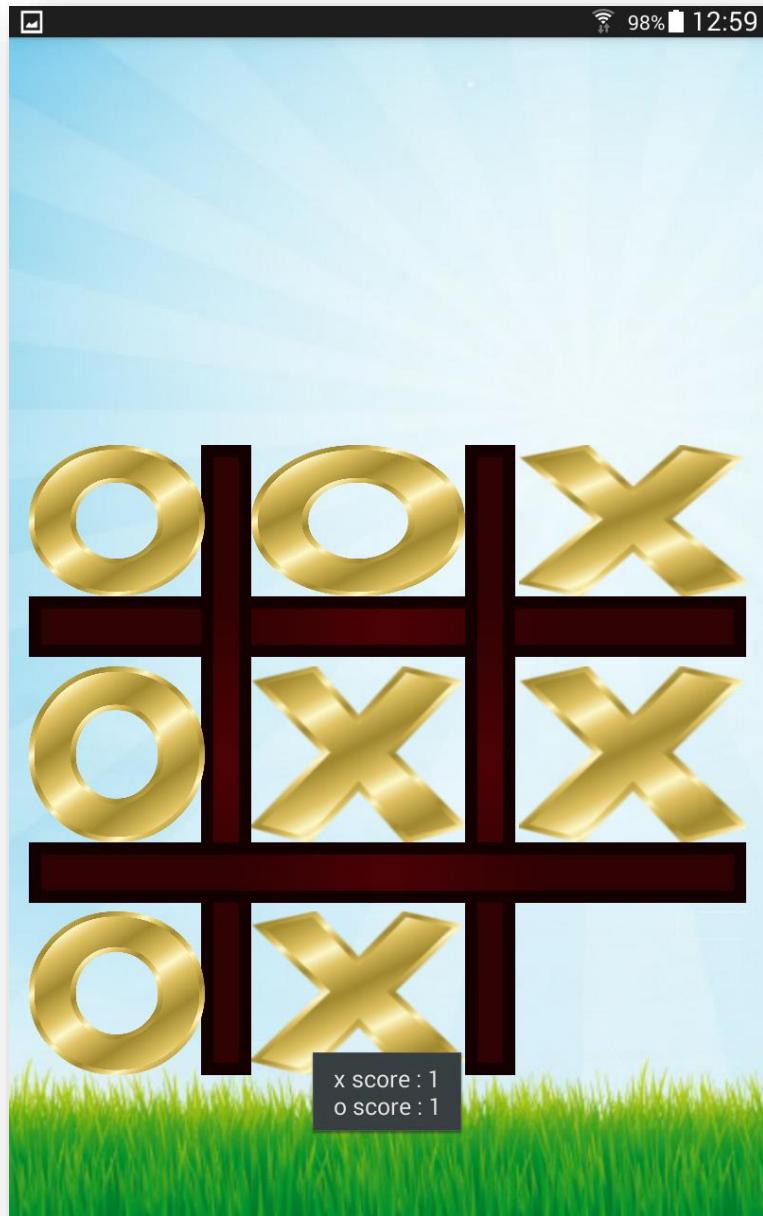


המשר O X

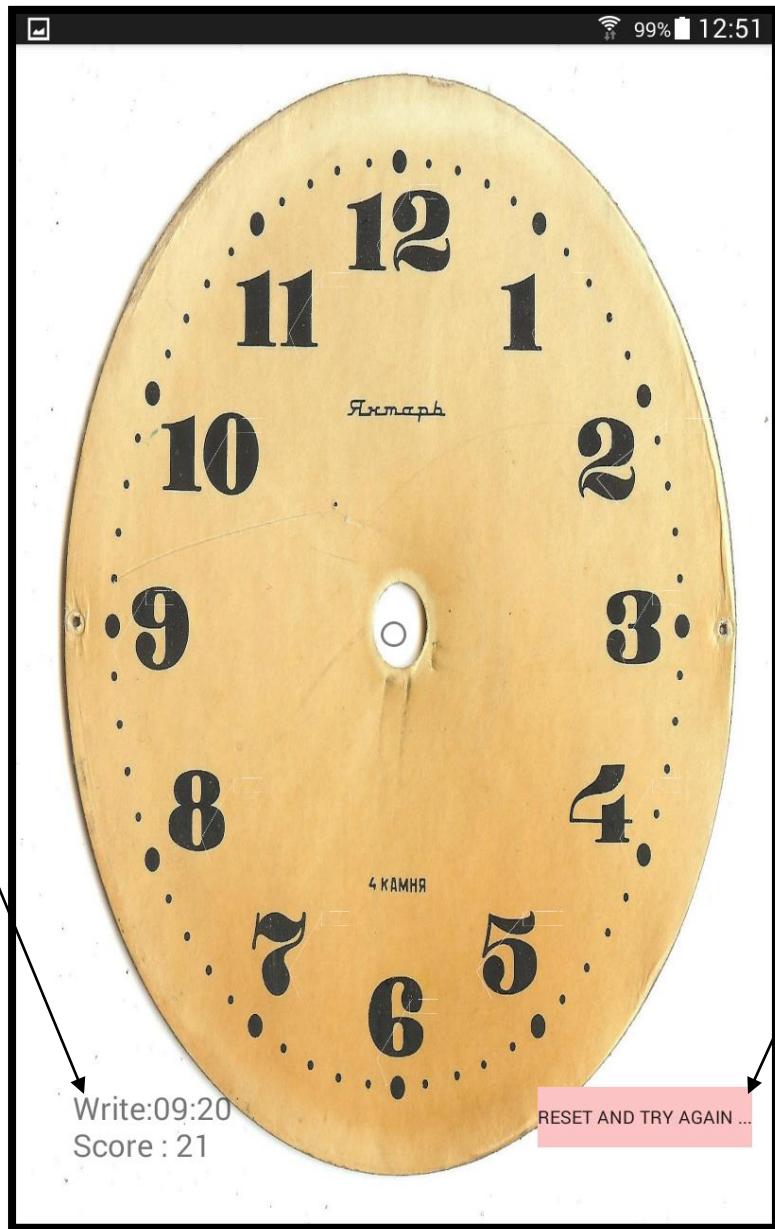
לבחור
משבצת
רצויה



המשר O X



CLOCK



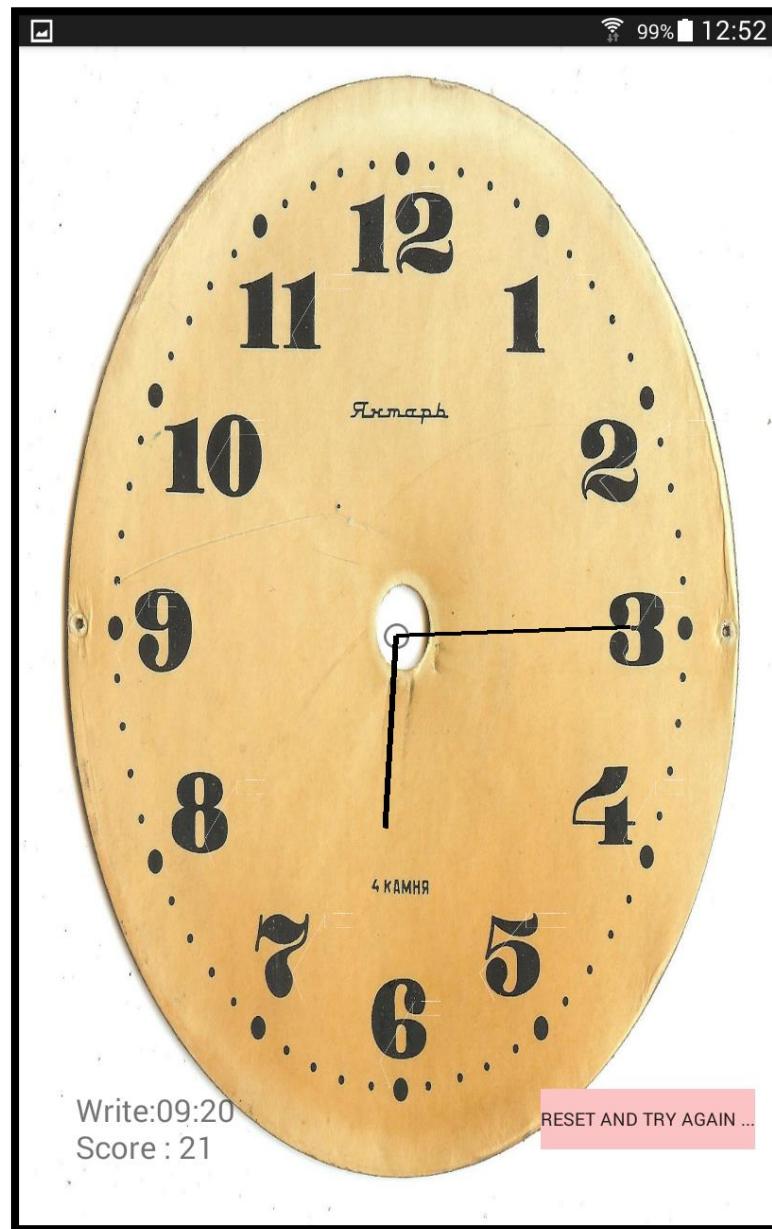
שעה
שכירים
לצייר וסוכם
נק'

משחק
השעון

איפוס
ולשחק
 מחדש

המשר CLOCK

מציריים קו
השעה אחר כר
קו הבדיקה אחר
כר ממחכים 2
שניות עד
שהאפליקציה י��יה
תבדוק את
התשובה שלנו
אם נכונה או לא



המשר CLOCK

כשלוחצים על כפתור
RESET AND TRY AGAIN
מאפסים את מה שיש ומצטירים
 מחדש

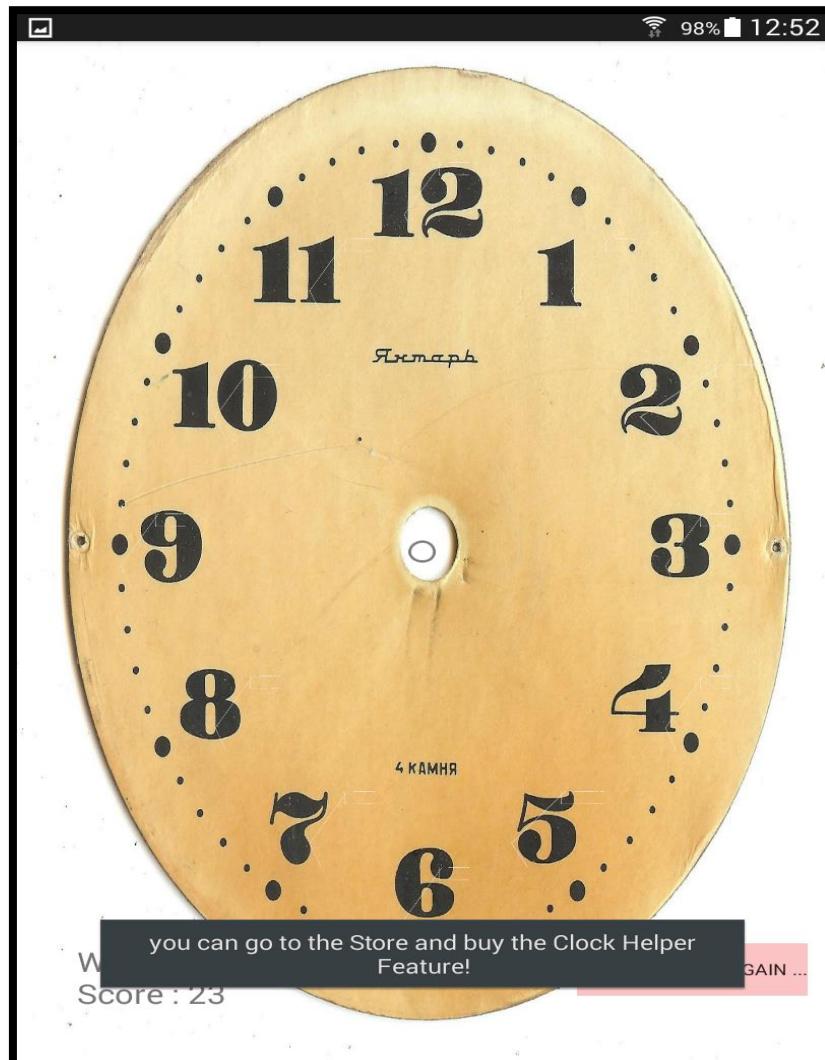


כשהתשובה שלנו לא נכונה אז
המשר רוד!!!!!!
(ימין/שמאל)

WRONG!!!!!!!

המשר CLOCK

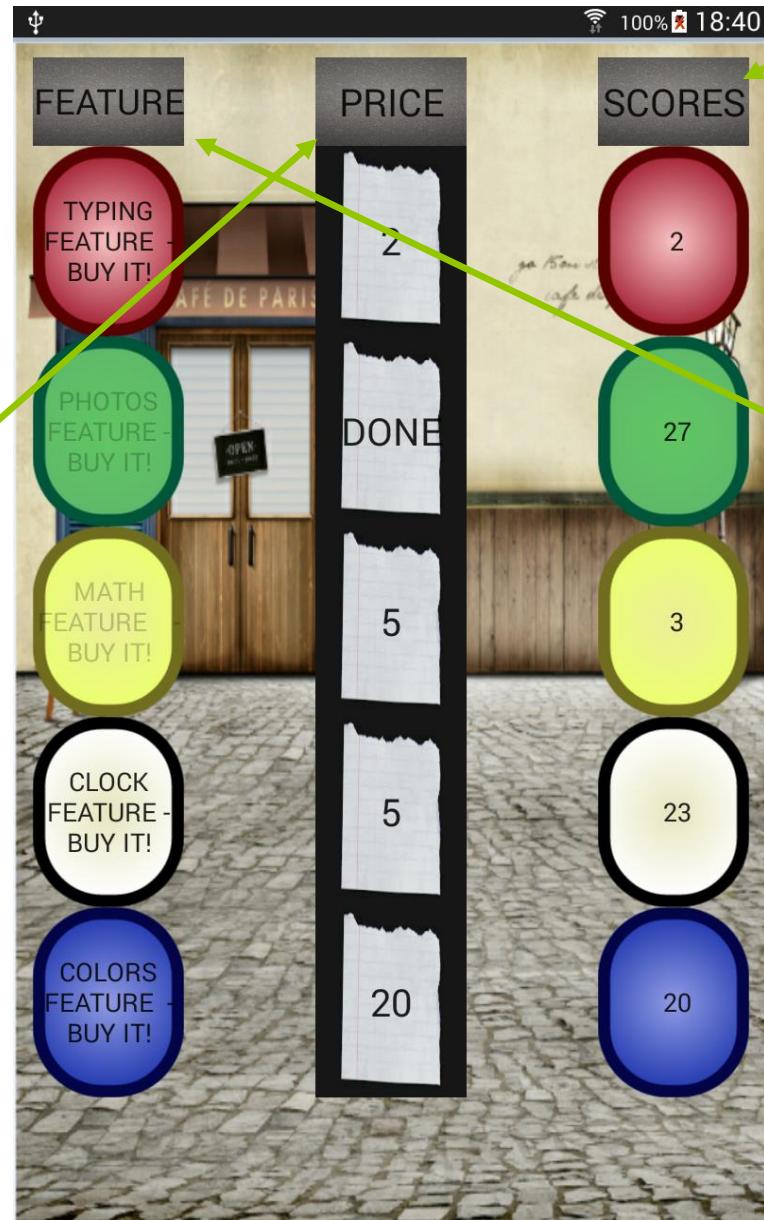
אם התשובה נכונה – צוברים נק' ואם
גם הגענו להטבה או תוצג לנו הודעה
"מגיע לכם הטבה!"



TAGU STORE

לכל
משחק יש
הטבות

PRICE
זה מס'
נקודות
שצריך
לצבור כדי
לקבל את
הטבה

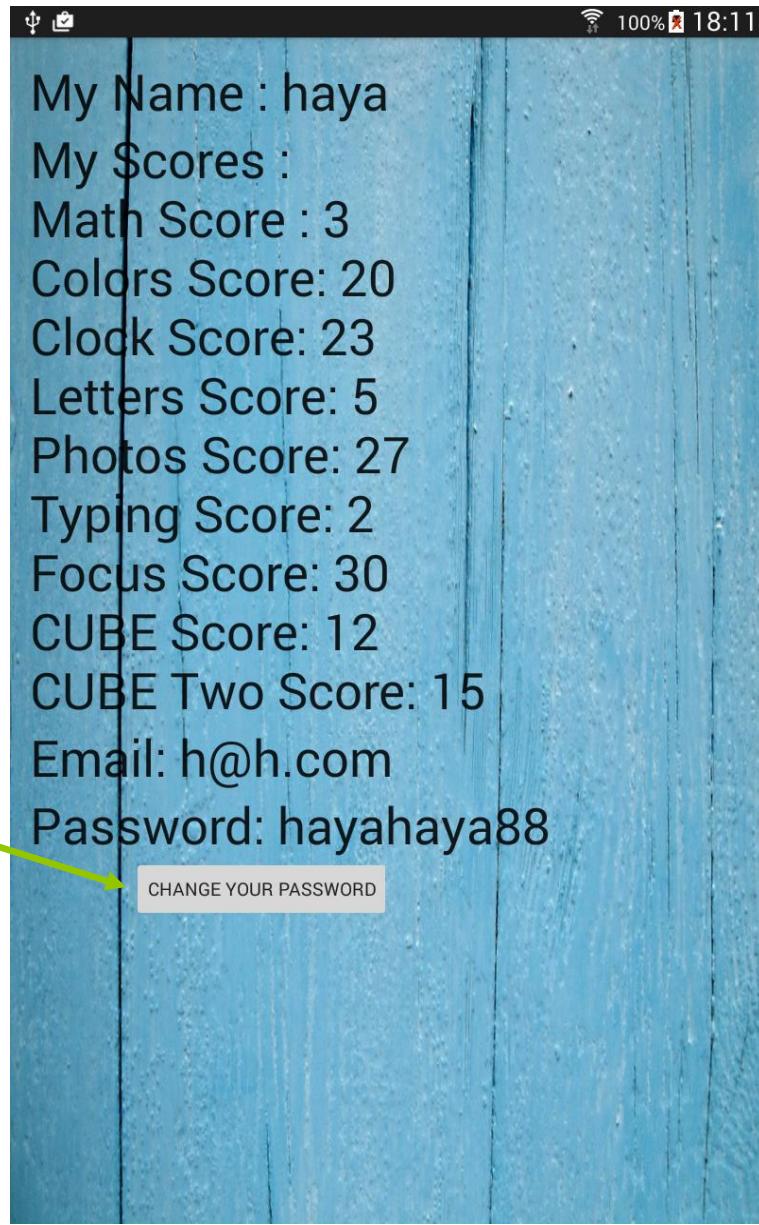


SCORES
מס'
נקודות של
משתמש

FEATURE
כפturny
קבלת
הטבה-
דליךim
ז"א שנוכל
לקבל
הטבה
עכשו

INFO

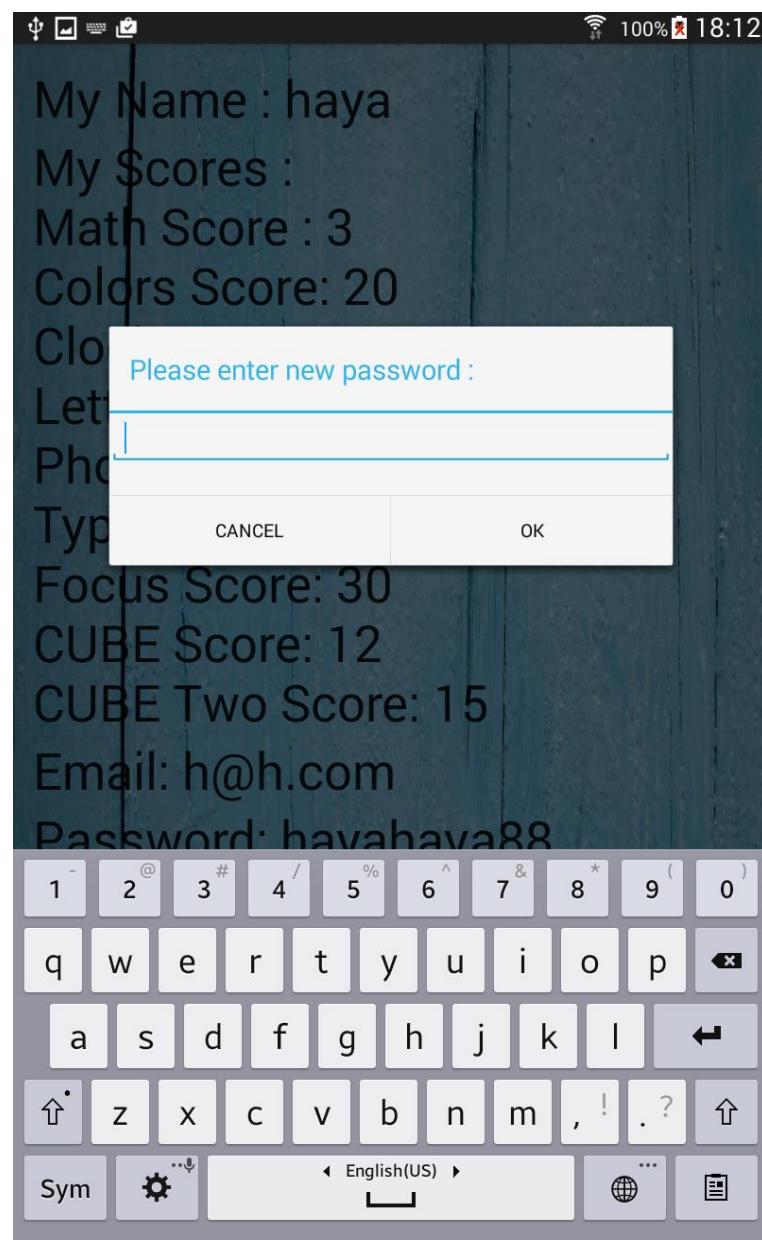
אפשרות
שינוי
סיסמה



דף שמציג
כל פרטי
המשתמש
ומס'
נקודות
שצבר בכל
משחק

המשר INFO

כשלוחצים על
CHANGE YOUR PASSOWRD
יופיע חלון לבחירת סיסמה חדשה



ABOUT



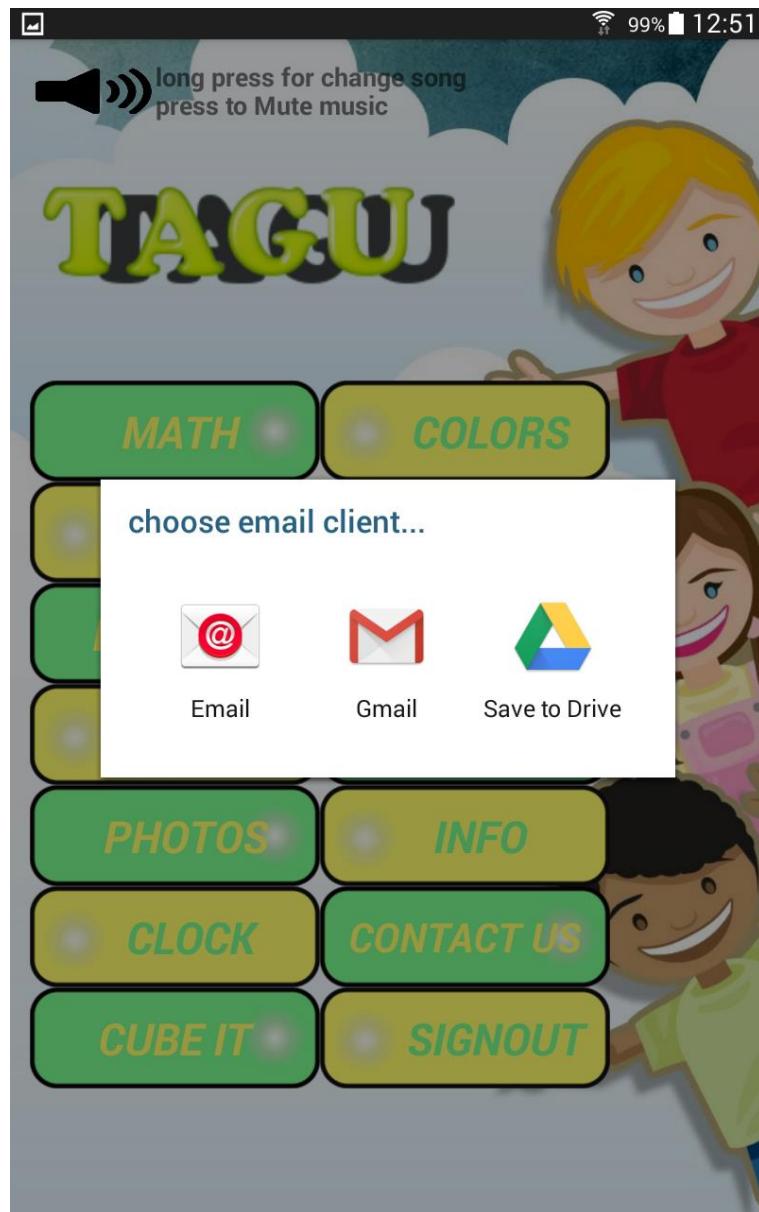
PAUSE THE
VIDEO

PLAY THE
VIDEO

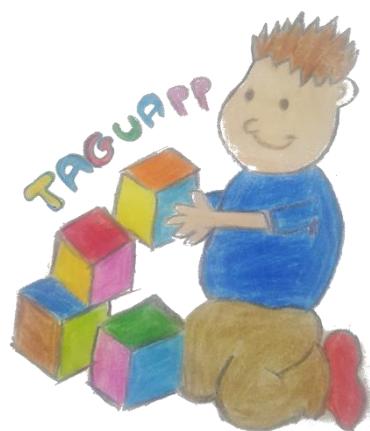
סרטון על
האפליקציה
וקצת הסבר.

CONTACT US

כשלוחצים על
CONTACT US
יופיע לנו חלון של כמה אפשרויות לצירור איתנו קשר



הסביר על DataBase



SQLite

קיימת טבלת Player ש מכילה את השדות הבאים :

תיאור שדה	שם שדה
מסוג INT מכיל שם משתמש	Name
מסוג TEXT מכיל סיסמה משתמש	Pass
math Score של letters	mScore
letters Score של typing	Letter
typing Score של clock	Typing
clock Score של colors	Clock
colors Score של photos	Colors
photos Score של focus	Photos
focus Score המכיל CLOCK INT	Focus
מכיל מס' הטעות העדכני של CLOCK INT	storeClock
מכיל מס' הטעות העדכני של TYPING INT	flagClock
מכיל Score מס' הטעות העדכני של TYPING INT	storeTyping
מכיל מס' הטעות העדכני של COLORS INT	flagTyping
מכיל Score מס' הטעות העדכני של COLORS INT	storeColors
מכיל מס' הטעות העדכני של MATH INT	flagColors
מכיל Score מס' הטעות העדכני של MATH INT	storeMath
מכיל מס' הטעות העדכני של PHOTOS INT	flagMath
מכיל Score מס' הטעות העדכני של PHOTOS INT	storePhotos
מכיל מס' הטעות העדכני של PHOTOS INT	flagPhotos
מסוג TEXT מכיל מייל של המשתמש	Mail

מסמך INT מכיל Score של 1	Cube
מסמך INT מכיל Score של 2	Cubetwo
מכיל רשימת מילים חדשות של TYPING TEXT	Arrlist

הערות :

1. שדה - Arrlist

מכיל מחרוזת שמורכבה ממספר מילים שהמשתמש מכניס בנוסף למילים שיש באפליקציה. יש אפשרות להכניס מילים חדשות במהלך ה הקלדה – typing.

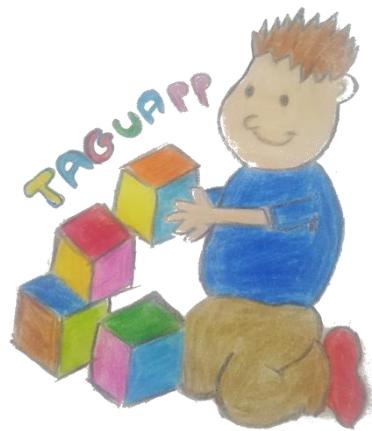
2. שדות ה – Flag

בודק לכל משתמש כמה הטעות מגיעו לפיה הנקודות שלו בכל משחק.

3. שדות ה – Store

לכל הטעה יש לה מס' נקודות מסוים ביחסות לצרכים להגעה אליו כדי לקבל את הטעה – כמשמעותם למס' נקודות זה התוצג לנו הודעה שmagieu לנו עכשו הטעה ומס' נקודות ביחסות מתעדכן למס' אחר ומכוון את עצמו להטעה הבאה.

פונקציות עיקריות



במחלקה *Signin*

isValidEmail(String email)

מетодה מקבלת מחרוזת ומחזירה אמת אם המחרוזת היא מייל אחרת מחזירה שקר.

ChangePassword()

אחרי בדיקת שם משתמש ומיל-אם נכונים המתודה שולחת סיסמה חדשה למשתמש במקרה של שכיחת סיסמה.

במחלקה *Math* משתמשים במס' פונקציות שמיינשנו במחלקה *MathClass*

ChangeRange()

מетодה רקורסיבית ש חוזרת על עצמה כל עוד טווח המספרים אינו בין 1 לבין המס' שנקלט או כל עוד התקבל קלט לא חוקי – צריך מספרים בלבד

locks(boolean bn)

מетодה מקבלת "שקר" נועלת כפטורី המשחק או "אמת" משחררת

CreateNumbers(int [] num,int con)

מетодה מקבלת מערך של 4 מספרים וטוויח שהתקבל ומיצרת לפחות תשובה נכון אחת כדי לוודא שניתן להרכיב ולהגיע לתוצאה מהמספרים המוצגים על המסך.

ShowResults(int [] num)

מетодה שמצויה לנו כל התשובות האפשריות.

Checking(int count,int index,char charr,int [] n)

מחזירה 1 או 3 או 4 אם התשובה נכונה

מחזירה 2 או 5 אם התשובה לא נכונה

מחזירה 0 או 6 אם עדין לא סיימנו את הנוסחה

ShowTheSecondCharacter(int [] num)

מетодה מקבלת מערך של מספרים שייצרנו ומחזירה סימן + או –
(כדי לדעת איזה סימן נכון לנוסחה – משתמשים בזה במקרה ואם יש לנו הטבת הבהוב)

ShowTheFirstCharacter(int [] num)

מетодה מקבלת מערך של מספרים שייצרנו ומחזירה את המס' הראשון (התו הראשון
בנוסחה) שמוגדר כחלק מהנוסחה הנכונה של האפליקציה ומשתמשים בזה במקרה
יש לנו הטבת הבהוב

ShowTheThirdCharacter(int [] num)

מетодה מקבלת מערך של מספרים שייצרנו ומחזירה את המס' השני (הतו השלישי בנוסחה) שmagdar כחלק מהנוסחה הנכונה של האפליקציה ומשתמשים בהז במקורה ואם יש לנו הטבת הבהוב

checkingthefirst(int s,Button num1,Button num2,Button num3,Button num4)

מетодה שמבצעיה את הבהוב ע"י אינמציה על הכפתורים שרשום בהם את המספרים שייצרנו.

checkingthesecound(char s,Button num5,Button num6)

מетодה שמבצעיה את הבהוב ע"י אינמציה על הכפתורים שרשום בהם את הסימן הנכון של הנוסחה (+ או -).

במחלקה typing

ADDAWORD()

מетодה שמוסיפה ושומרת במסד נתונים עוד מילים חדשות של משתמש ווחזרת על עצמה באופן רקורסיבי כל עוד המשתמש מכניס עוד מילים.

במחלקה letters

SpeakerFunction(final String str)

מетодה מקבלת מחרוזת ומפעילה עליה הדיבור וניתן לשמעו את המחרוזת.

AnimateandSlideShow()

מетодה שמבצעיה אינמציה על הצגת האותיות וגם מגבילה את טווח האותיות שמצווגות כך שלא ייקח הרבה זמן עד שהאות תוצג לנו

במחלקה cube Level 1

RandomllyRunner()

מетодה מרכזית שמופעלת כל הזמן שבוחרת באופן אקראי איזו קובייה עכשו תשחק ותופעל ובאיזה צבע באופן אקראי

changerColorsButton(Button b)

מетодה שבוחרת באופן אקראי צבע לקובייה (4 צבעים)

SlideToDown()

מетодה שמצויה את הקובייה למטה ובודקת אחרי ההתגשות אם התנגשה עם קובייה RandomllyRunner()
שיש לה אותו צבע אם כן צוברת עוד נק' ומפעילה מחדש את ()

SlideToAbove()

מетодה שמצויה את הקubiיה למעלה ובודקת אחרי ההתגשות אם התנגשה עם קובייה RandomllyRunner()
שיש לה אותו צבע אם כן צוברת עוד נק' ומפעילה מחדש את ()

chancepositions()

מетодה שמשנה את מקום הקוביות שבאמצעו ע"מ להתאים אותן עם צבע הקובייה
שמגיעה מלמעלה או מלמטה.

במחלקה cube Level 2

RandomllyRunner()

מетодה מרכזית שמופעלת כל הזמן שבוחרת באופן אקראי איזו קוביות עכשו תשחקנה
ותופעלה ובאיזה צבע באופן אקראי

changerpositionTwo()

מетодה שמשנה את מקום הקוביות שבצד השני ע"מ להתאים אותן עם צבע הקובייה
שמגיעה מלמעלה או מלמטה.

changerpositionOne()

מетодה שמשנה את מקום הקוביות שבצד הראשון ע"מ להתאים אותן עם צבע הקובייה
שמגיעה מלמעלה או מלמטה.

LeftKeysSwitcher()

מетодה משנה את המפתחים של הקוביות מצד השני – אם קובייה לבנה למטה הופכת
אותה למעלה ולהפוך ואם קובייה צהובה למטה הופכת אותה למעלה ולהפוך

RightKeysSwitcher()

מетодה משנה את המפתחים של הקוביות מצד הראשון – אם קובייה לבנה למטה הופכת
אותה למעלה ולהפוך ואם קובייה צהובה למטה הופכת אותה למעלה ולהפוך

changerColorsBallTwo(Button b)

מетодה שבוחרת באופן אקראי צבע לקוביות הצד השני (שמאל) – 2 צבעים

changerColorsBallOne(Button b)

מетодה שבוחרת באופן אקראי צבע לקוביות הצד הראשון (ימני) – 2 צבעים

SlideToDown()

מетодה שמצויה את הקוביות למטרה ובודקת אחרי ההתנגשות אם התנגשו עם קוביות שיש להן אותו צבע (לכל קובייה בהתאם עם הצד שלה) אם כן צוברת עוד נק' ומפעילה מחדש את RandomllyRunner()

SlideToAbove()

מетодה שמצויה את הקוביות למעלה ובודקת אחרי ההתנגשות אם התנגשו עם קוביות שיש להן אותו צבע (לכל קובייה בהתאם עם הצד שלה) אם כן צוברת עוד נק' ומפעילה מחדש את RandomllyRunner()

במחלקה clock משתמשים בממ' פונקציות שמיינשנו

ClockLine And MinuteLine

locks(boolean bn)

מетодה מקבלת "שקר" נועלת כפטורי המשחק או "אמת" משחררת

onDraw(Canvas canvas)

מетодה שמצוירת קו עם אורך מסוים – קו שעלה שונה באורכו מקו דקה

onFinish() Function – when the time is done

מетодה שמופעלת אחרי שהזמן נגמר ובודקת אם ציררנו את השעה עם הדקות בצורה נכונה.

במחלקה TAGUAPP

playMusic(View view)

מетодה שמבצעיה/עוצרת את המוזיקה כשלוחצים על כפתור השמע.

Drag/Drop to swap buttons positions option by a DragListener Class

אפשריה להחליף את המיקומות של הcptoris בדף הראשי כך כלוחצים לחיצה ארוכה על כפתור אז שומרים במשתנה עזר את מקומו ושמים אותו במקום כפתור שני ואז מחליפים את מקומות של שני הcptoris.

מקום כפתור מוגדר :

TOP – BOTTOM – LEFT – RIGHT של כל כפתור.

במחלקה my_db

add_player(String name , String pass, String mail)

מетодה שמօסיפה משתמש חדש ומתחילה לו אפס נק' למשחק המתמטיקה במידה והמשתמש לא קיימ.

updateCurrentPassword(String name, String pass)

מетодה שמעדכנת את הסיסמה – במקורה של שינוי סיסמה.

getPassword(String str1, String str2)

מетодה שמחזירה את הסיסמה של המשתמש.

getEmail(String str1, String str2)

מетодה שמחזירה את המail של המשתמש.

searchSignUp(String str)

מетодה מחזירה "אמת" אם המשתמש כבר קיים במערכת – אחרת מחזירה "שקר".

searchEmailSignUp(String mail)

מетодה מחזירה "אמת" אם מייל המשתמש כבר קיים במערכת – אחרת מחזירה "שקר".

searchSignIn(String str1, String str2)

מетодה מחזירה "אמת" אם המשתמש קיים במערכת – אחרת מחזירה "שקר".

searchforgotyourpassword(String user, String mail)

מетодה שמחזירה "אמת" אם המשתמש ו对他-mail שלו קיימים במערכת – אחרת שמחזירה "שקר".

get_user_name(String s)

מетодה שמחזירה שם המשתמש.

במחלקה UserInfo

PasswordChanger()

מетодה רקורסיבית שמשנה את הסיסמה.

במחלקה OX

StartWork()

מетодה שמפסיק ומנקה את המסך ומcinha את המשחק למשחק חדש.

locks(boolean lock)

מетодה מקבלת "שקר" נועלת כפטורי המשחק | "אמת" משחררת

setrandom()

מетодה שבודקת אם יש מצב שהאפליקציה קציה לנץ או אם יש מצב שתימנע מהמשתמש לנץ – אם שני המצבים לא מתקינים אז האפליקציה תשים איקס/עיגול במקום ריק שלא מפריע.

checker()

מетодה שבודקת מצב המשחק אחרי הצבת איקס או עיגול אם יש לנו לנץ – אם לא ממשיכים לשחק.

setplace(int sog , int index,Button c)

מетодה שמציבה איקס או עיגול במקום מסוים.

check(int index)

מетодה מחזירה "אמת" אם המקום שמתקיים ריק – אחרת מחזירה "שקר"

Photos

b1(View v)

מетодה שקשורה לכפטור הראשון ובודקת אם בו רשומה את התשובה הנכונה או לא ובכל
מקרה בוחרים מילים חדשות עם הצגת תמונה חדשה וועברים לשלב הבא

b2(View v)

מетодה שקשורה לכפטור השני ובודקת אם בו רשומה את התשובה הנכונה או לא ובכל
מקרה בוחרים מילים חדשות עם הצגת תמונה חדשה וועברים לשלב הבא

b3(View v)

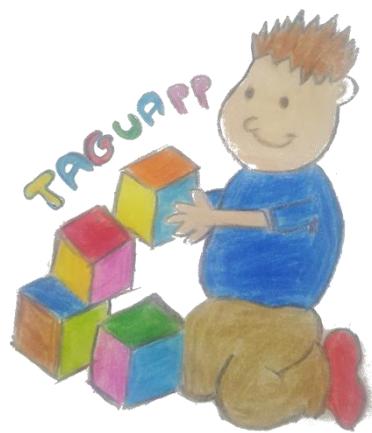
מетодה שקשורה לכפטור השלישי ובודקת אם בו רשומה את התשובה הנכונה או לא ובכל
מקרה בוחרים מילים חדשות עם הצגת תמונה חדשה וועברים לשלב הבא

Focus

check(int x)

מетодה שבודקת את המקום שהתקבל הוא המקום הנכון או לא.

קוד והסבירים



קובץ XML

קובץ activity_welcome.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context="jof.jof.welcome"
    android:background="@drawable/back1">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="match_parent"
        android:id="@+id/WelcomePhoto"
        android:layout_marginTop="135dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:background="@mipmap/best" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="WELCOME TO TAGUAPP"
        android:id="@+id/welcometxt"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="47dp"
        android:textSize="30dp"
        android:textColor="#ff0004" />
</RelativeLayout>
```

קובץ rotate.xml

קובץ עזר לאנימציה

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <rotate
        android:duration="3500"
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%">
    </rotate>
</set>
```

קובץ activity_signin.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.Signin"
    android:background="@drawable/back3">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Name :"
        android:id="@+id/txtName"
        android:layout_above="@+id/etName"
        android:layout_alignLeft="@+id/etName"
        android:layout_alignStart="@+id/etName" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Password :"
        android:id="@+id/txtPassword"
        android:layout_below="@+id/textView5"
        android:layout_toLeftOf="@+id/txtnewname"
        android:layout_toStartOf="@+id/txtnewname" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="New Name :"
        android:id="@+id/txtnewname"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginRight="259dp"
        android:layout_marginEnd="259dp"
        android:visibility="invisible" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="New Password :"
        android:id="@+id/txtnewpassword"
        android:layout_above="@+id/etnewpassword"
        android:layout_alignLeft="@+id/btnSignIn"
        android:layout_alignStart="@+id/btnSignIn" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign Up"
        android:id="@+id/btnSignUp"
```

```

        android:background="@drawable/colorbutton"
        android:layout_alignParentBottom="true"
        android:layout_alignLeft="@+id/etnewpassword"
        android:layout_alignStart="@+id/etnewpassword" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/etnewpassword"
    android:layout_above="@+id/btnSignUp"
    android:layout_alignLeft="@+id/txtnewpassword"
    android:layout_alignStart="@+id/txtnewpassword" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/etName"
    android:layout_above="@+id/txtPassword"
    android:layout_toLeftOf="@+id/checkpass"
    android:layout_toStartOf="@+id/checkpass" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sing in"
    android:id="@+id/btnSignIn"
    android:background="@drawable/colorbutton"
    android:layout_centerVertical="true"
    android:layout_alignLeft="@+id/txtforgoturpass"
    android:layout_alignStart="@+id/txtforgoturpass" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="New Name :"
    android:id="@+id/textView"
    android:layout_above="@+id/etnewname"
    android:layout_alignLeft="@+id/etnewname"
    android:layout_alignStart="@+id/etnewname" />

<ImageView
    android:layout_width="350dp"
    android:layout_height="130dp"
    android:id="@+id/imageView4"
    android:background="@drawable/cooltext172153282331985"
    android:layout_alignTop="@+id/txtnewname"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageView9"
    android:background="@drawable/clienticon"
    android:layout_below="@+id/imageView4"
    android:layout_alignRight="@+id/textView"
    android:layout_alignEnd="@+id/textView" />

```

```

<TextView
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Think And Grow Up"
    android:id="@+id/textView5"
    android:textSize="35dp"
    android:textColor="#000000"
    android:textStyle="bold"
    android:background="@drawable/abc_list_selector_disabled_holo_light"
    android:layout_below="@+id/imageView4" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/checkpass"
    android:layout_alignBaseline="@+id/etnewpassword"
    android:layout_alignBottom="@+id/etnewpassword"
    android:layout_toRightOf="@+id/etnewpassword"
    android:layout_toEndOf="@+id/etnewpassword"
    android:layout_marginLeft="43dp"
    android:layout_marginStart="43dp"
    android:textSize="25dp"
    android:textStyle="bold|italic" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/etpassword"
    android:layout_below="@+id/txtPassword"
    android:layout_alignLeft="@+id/txtPassword"
    android:layout_alignStart="@+id/txtPassword" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/etnewname"
    android:layout_above="@+id/txtnewpassword"
    android:layout_toLeftOf="@+id/checkpass"
    android:layout_toStartOf="@+id/checkpass"
    android:layout_marginBottom="21dp" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:ems="10"
    android:id="@+id/etyourEmail"
    android:layout_above="@+id/textView"
    android:layout_alignLeft="@+id/textView"
    android:layout_alignStart="@+id/textView"
    android:layout_marginBottom="21dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Your Email"

```

```
        android:id="@+id/urMail"
        android:layout_above="@+id/etyourEmail"
        android:layout_alignLeft="@+id/etyourEmail"
        android:layout_alignStart="@+id/etyourEmail" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Forgot your password?"
        android:id="@+id/txtforgoturpass"
        android:textStyle="italic"
        android:singleLine="false"
        android:textColor="#0073ff"
        android:layout_below="@+id/etpassword"
        android:layout_alignRight="@+id/etpassword"
        android:layout_alignEnd="@+id/etpassword" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="New User?"
        android:id="@+id/newusername"
        android:layout_above="@+id/urMail"
        android:layout_alignLeft="@+id/urMail"
        android:layout_alignStart="@+id/urMail"
        android:layout_marginBottom="63dp"
        android:textSize="30dp"
        android:textStyle="italic" />
</RelativeLayout>
```

קובץ activity_taguapp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.TAGUAPP"
    android:background="@drawable/yes"
    android:screenOrientation="portrait">

    <ImageView
        android:layout_width="350dp"
        android:layout_height="100dp"
        android:id="@+id/imageView"
        android:background="@drawable/cooltext172153282331985"
        android:layout_marginTop="40dp"
        android:layout_below="@+id/sound"
        android:layout_alignRight="@+id/textView9"
        android:layout_alignEnd="@+id/textView9" />

    <Button
        android:layout_width="225dp"
        android:layout_height="80dp"
        android:id="@+id/math"
        android:background="@drawable/rect"
        android:text=" MATH "
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#ffffd6b"
        android:typeface="normal"
        android:layout_alignTop="@+id/btnco"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="225dp"
        android:layout_height="80dp"
        android:id="@+id/typing"
        android:background="@drawable/rect1"
        android:text=" Typing"
        android:textAlignment="center"
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#7afa91"
        android:typeface="normal"
        android:layout_alignTop="@+id/btnXO"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="80dp"
        android:id="@+id/letter"
        android:background="@drawable/rect"
        android:text=" LETTERS"
```

```
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#ffffd6b"
        android:typeface="normal"
        android:layout_below="@+id/typing"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignRight="@+id/typing"
        android:layout_alignEnd="@+id/typing" />

<Button
    android:layout_width="250dp"
    android:layout_height="80dp"
    android:id="@+id/focus1"
    android:background="@drawable/rect1"
    android:text=" FOCUS "
    android:textAlignment="center"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#7afa91"
    android:typeface="normal"
    android:layout_below="@+id/letter"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toLeftOf="@+id/about"
    android:layout_toStartOf="@+id/about" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:id="@+id/photos1"
    android:background="@drawable/rect"
    android:text=" Photos"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#ffffd6b"
    android:typeface="normal"
    android:layout_below="@+id/focus1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toLeftOf="@+id/btnco"
    android:layout_toStartOf="@+id/btnco" />

<Button
    android:layout_width="250dp"
    android:layout_height="80dp"
    android:id="@+id/centers"
    android:background="@drawable/rect1"
    android:text=" CLOCK"
    android:textAlignment="center"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#7afa91"
    android:typeface="normal"
    android:layout_below="@+id/photos1"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toLeftOf="@+id/contactus"
    android:layout_toStartOf="@+id/contactus" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
```

```
    android:id="@+id/cubing"
    android:background="@drawable/rect"
    android:text="Cube it"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#fffd6b"
    android:typeface="normal"
    android:layout_alignTop="@+id/signout"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toLeftOf="@+id/signout"
    android:layout_toStartOf="@+id/signout" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:id="@+id/btnXO"
    android:background="@drawable/rect"
    android:textAlignment="center"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#fffd6b"
    android:typeface="normal"
    android:layout_below="@+id/math"
    android:layout_toRightOf="@+id/typing"
    android:layout_alignRight="@+id/btnco"
    android:layout_alignEnd="@+id/btnco"
    android:text="X O" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:id="@+id/about"
    android:background="@drawable/rect"
    android:text="ABOUT"
    android:textSize="35dp"
    android:textStyle="bold|italic"
    android:textColor="#fffd6b"
    android:typeface="normal"
    android:layout_above="@+id/photos1"
    android:layout_toRightOf="@+id/photos1"
    android:layout_alignRight="@+id/userinfo"
    android:layout_alignEnd="@+id/userinfo" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:id="@+id/contactus"
    android:background="@drawable/rect"
    android:text="CONTACT US"
    android:textAlignment="center"
    android:textSize="32dp"
    android:textStyle="bold|italic"
    android:textColor="#fffd6b"
    android:typeface="normal"
    android:layout_alignTop="@+id/centers"
    android:layout_alignLeft="@+id/about"
    android:layout_alignStart="@+id/about"
    android:layout_alignRight="@+id/btnco"
    android:layout_alignEnd="@+id/btnco" />

<Button
    android:layout_width="225dp"
```

```

        android:layout_height="80dp"
        android:id="@+id/btnco"
        android:background="@drawable/rect1"
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#7afa91"
        android:typeface="normal"
        android:text="    Colors"
        android:layout_below="@+id/imageView"
        android:layout_toRightOf="@+id/math"
        android:layout_toEndOf="@+id/math"
        android:layout_marginTop="65dp" />

<Button
        android:layout_width="225dp"
        android:layout_height="80dp"
        android:id="@+id/btnStore"
        android:background="@drawable/rect1"
        android:textAlignment="center"
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#7afa91"
        android:typeface="normal"
        android:layout_below="@+id/typing"
        android:layout_toRightOf="@+id/typing"
        android:layout_toEndOf="@+id/typing"
        android:text="TAGU Store" />

<Button
        android:layout_width="225dp"
        android:layout_height="80dp"
        android:id="@+id/signout"
        android:background="@drawable/rect1"
        android:text="    SignOut"
        android:textAlignment="center"
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#7afa91"
        android:typeface="normal"
        android:layout_below="@+id/contactus"
        android:layout_alignLeft="@+id/contactus"
        android:layout_alignStart="@+id/contactus" />

<Button
        android:layout_width="225dp"
        android:layout_height="80dp"
        android:id="@+id/userinfo"
        android:background="@drawable/rect1"
        android:text="    INFO"
        android:textAlignment="center"
        android:textSize="35dp"
        android:textStyle="bold|italic"
        android:textColor="#7afa91"
        android:typeface="normal"
        android:layout_below="@+id/focus1"
        android:layout_toRightOf="@+id/photos1"
        android:layout_toEndOf="@+id/photos1" />

<Button
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:id="@+id/sound"
        android:background="@drawable/startmusic"

```

```
        android:onClick="playMusic"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="long press for change song"
        android:id="@+id/textView9"
        android:layout_alignTop="@+id/sound"
        android:layout_toRightOf="@+id/sound"
        android:layout_toEndOf="@+id/sound"
        android:textSize="20dp"
        android:textStyle="bold" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="press to Mute music"
        android:id="@+id/textView10"
        android:textSize="20dp"
        android:layout_above="@+id/imageView"
        android:layout_toRightOf="@+id/sound"
        android:layout_toEndOf="@+id/sound"
        android:textStyle="bold" />

</RelativeLayout>
```

קובץ activity_math.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.Math"
    android:background="@drawable/reka2">

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/bbb1"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_above="@+id/b5"
        android:layout_alignLeft="@+id/txtresult"
        android:layout_alignStart="@+id/txtresult" />

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/b22"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_below="@+id/txtresult"
        android:layout_toRightOf="@+id/bbb1"
        android:layout_toEndOf="@+id/bbb1" />

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/b3"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_alignTop="@+id/b22"
        android:layout_toRightOf="@+id/b22"
        android:layout_toEndOf="@+id/b22" />

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/b4"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_below="@+id/bbb1"
        android:layout_toLeftOf="@+id/b22"
        android:layout_toStartOf="@+id/b22" />

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/b5"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_alignTop="@+id/b22"
        android:layout_alignLeft="@+id/b3"
        android:layout_alignStart="@+id/b3" />
```

```

        android:id="@+id/b5"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_below="@+id/b22"
        android:layout_alignLeft="@+id/b22"
        android:layout_alignStart="@+id/b22" />

    <Button
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:id="@+id/b6"
        android:background="@drawable/math1"
        android:onClick="gene"
        android:textSize="40dp"
        android:layout_below="@+id/b22"
        android:layout_toRightOf="@+id/b22"
        android:layout_toEndOf="@+id/b22" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/txtresult"
        android:textSize="35dp"
        android:text="find the equation that equals ="
        android:textColor="#095e15"
        android:hint="tow numbers from matrix that equals this number"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/txtchangerange"
        android:layout_toEndOf="@+id/txtchangerange"
        android:layout_marginLeft="68dp"
        android:layout_marginStart="68dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnagain"
        android:onClick="gene"
        android:background="@drawable/clienticon"
        android:layout_above="@+id/textView6"
        android:layout_alignRight="@+id/textView6"
        android:layout_alignEnd="@+id/textView6" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/Clear"
        android:layout_gravity="bottom"
        android:text="C"
        android:background="@drawable/math2"
        android:layout_below="@+id/b5"
        android:layout_toRightOf="@+id/b5"
        android:layout_toEndOf="@+id/b5" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/txtScore"
        android:textSize="35dp"
        android:text="score"
        android:textColor="#095e15"

```

```

        android:layout_alignParentTop="true"
        android:layout_toLeftOf="@+id/txtresult"
        android:layout_toStartOf="@+id/txtresult" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:id="@+id/txtres"
    android:textSize="25dp"
    android:layout_below="@+id/Clear"
    android:layout_toRightOf="@+id/txtchangerange"
    android:layout_toEndOf="@+id/txtchangerange" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/txtnos7a2"
    android:textSize="30dp"
    android:layout_above="@+id/txtres"
    android:layout_below="@+id/b4"
    android:layout_alignLeft="@+id/txtres"
    android:layout_alignStart="@+id/txtres"
    android:layout_toLeftOf="@+id/b5"
    android:layout_toStartOf="@+id/b5" />

<TextView
    android:layout_width="200dp"
    android:layout_height="370dp"
    android:text="Hello, That's me TAGU , if you dont know to solve This just click
me."
    android:id="@+id/textView6"
    android:textSize="25dp"
    android:background="@drawable/abc_list_selector_disabled_holo_light"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/txtresult"
    android:layout_alignEnd="@+id/txtresult" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Change the range"
    android:id="@+id/txtchangerange"
    android:textSize="20dp"
    android:textColor="#bb0000"
    android:layout_alignBottom="@+id/textView6"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>

```

קובץ slide_down_out.xml

קובץ עוזר לאנימציה של הצגת הנוסחה במשחק המתמטיקה

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="0.0"
        android:toXScale="1.0"
        android:toYScale="1.0"/>
</set>
```

קובץ activity_typering.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.typering"
    android:background="@drawable/typering111">

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/etType"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="151dp"
        android:gravity="center" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/CUBESCORE"
        android:textSize="35dp"
        android:layout_above="@+id/b"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/b"
        android:background="@drawable/colorbutton"
        android:textColor="#dc19287d"
        android:textSize="30dp"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="false"
        android:layout_alignParentRight="false"
        android:layout_alignParentEnd="false" />

    <Button
        android:layout_width="90dp"
        android:layout_height="fill_parent"
        android:id="@+id/button2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/typeringf"
        android:layout_alignParentBottom="true" />

    <Button
        android:layout_width="90dp"
        android:layout_height="fill_parent"
        android:id="@+id/button3"
        android:background="@drawable/typeringf"
```

```
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true" />

    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/addwords"
        android:layout_above="@+id/CUBESCORE"
        android:layout_centerHorizontal="true"
        android:text="ADD NEW WORDS"
        android:background="#fbc4c4" />
</RelativeLayout>
```

קובץ activity_letters.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.letters"
    android:background="@drawable/letters222">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="350dp"
        android:id="@+id/ivletters"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Catch the letter : "
        android:id="@+id/txtcatch"
        android:textSize="40dp"
        android:layout_below="@+id/ivletters"
        android:layout_centerHorizontal="true"
        android:background="@drawable/abc_list_selector_disabled_holo_dark" />

    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Hello , I am TAGU , if you dont remind how to expel the slide
letter just click me."
        android:id="@+id/textView3"
        android:background="@drawable/abc_list_selector_disabled_holo_dark"
        android:layout_alignBottom="@+id/txtcatch"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnHomePage"
        android:background="@drawable/clienticon"
        android:layout_above="@+id/textView3"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:id="@+id/wife"
        android:background="@drawable/taguwife"
        android:layout_above="@+id/textView4"
        android:layout_alignParentRight="true"
```

```
        android:layout_alignParentEnd="true" />

<TextView
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Hello , I am TAGU wife , i will help you to expel the written
letter."
    android:id="@+id/textView4"
    android:background="@drawable/abc_list_selector_disabled_holo_dark"
    android:layout_alignTop="@+id/textView3"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Change a letter"
    android:id="@+id/chngletter"
    android:background="#fbc4c4"
    android:layout_below="@+id/txtcatch"
    android:layout_alignRight="@+id/ivletters"
    android:layout_alignEnd="@+id/ivletters" />
</RelativeLayout>
```

קובץ activity_focus.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".focus"
    android:background="@drawable/focus12">

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr3"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr2"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr1"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr6"
        android:layout_above="@+id/rr2"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="119dp" />

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr5"
        android:layout_alignTop="@+id/rr6"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:id="@+id/rr4"
        android:layout_alignTop="@+id/rr6"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
```

```
<Button
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:id="@+id/rr7"
    android:layout_alignTop="@+id/rr10"
    android:layout_alignLeft="@+id/rr6"
    android:layout_alignStart="@+id/rr6" />

<Button
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:id="@+id/rr8"
    android:layout_alignTop="@+id/rr10"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<Button
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:id="@+id/rr9"
    android:layout_below="@+id/button4"
    android:layout_alignLeft="@+id/rr7"
    android:layout_alignStart="@+id/rr7" />

<Button
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:id="@+id/rr10"
    android:layout_above="@+id/rr4"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_marginBottom="93dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button4"
    android:background="@drawable/abc_popup_background_mtrl_mult"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textSize="35dp" />
</RelativeLayout>
```

קובץ activity_photoss.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.photoss"
    android:background="@drawable/stst">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/chance1"
        android:onClick="b1"
        android:background="#7f94da"
        android:textStyle="bold"
        android:layout_alignParentBottom="true"
        android:layout_alignLeft="@+id/imageView2"
        android:layout_alignStart="@+id/imageView2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/chance2"
        android:onClick="b2"
        android:background="#7f94da"
        android:textStyle="bold"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/chance3"
        android:onClick="b3"
        android:background="#7f94da"
        android:textStyle="bold"
        android:layout_alignParentBottom="true"
        android:layout_alignRight="@+id/imageView2"
        android:layout_alignEnd="@+id/imageView2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Score : "
        android:id="@+id/score5"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/timerphoto"
```

```
    android:layout_below="@+id/score5"
    android:layout_marginTop="20dp"
    android:background="#ff0000"
    android:layout_alignRight="@+id/score5"
    android:layout_alignEnd="@+id/score5" />

<ImageView
    android:layout_width="400dp"
    android:layout_height="320dp"
    android:id="@+id/imageView2"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />

</RelativeLayout>
```

קובץ activity_cubeit.xml

LEVEL 1

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back2">

    <Button
        android:id="@+id/movetodown"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_alignParentTop="true"
        android:layout_alignLeft="@+id/movetoabove"
        android:layout_alignStart="@+id/movetoabove"
        android:layout_alignRight="@+id/movetoabove"
        android:layout_alignEnd="@+id/movetoabove"
        android:textColor="#ff0000" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/bluemain"
        android:background="#0008ff"
        android:layout_above="@+id/switcher"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/switcher"
        android:layout_centerVertical="true"
        android:layout_alignLeft="@+id/bluemain"
        android:layout_alignStart="@+id/bluemain"
        android:background="@drawable/rotatee" />

    <Button
        android:id="@+id/movetoabove"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:textColor="#ff0000" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/blackmain"
        android:background="#000000"
        android:layout_below="@+id/bluemain"
        android:layout_toLeftOf="@+id/bluemain"
        android:layout_toStartOf="@+id/bluemain" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:background="#ff0015"
```

```
        android:layout_alignBottom="@+id/switcher"
        android:layout_toRightOf="@+id/switcher"
        android:layout_toEndOf="@+id/switcher"
        android:id="@+id/redmain" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/greenmain"
        android:background="#26ff00"
        android:layout_below="@+id/switcher"
        android:layout_alignLeft="@+id/switcher"
        android:layout_alignStart="@+id/switcher" />

    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="level 2? "
        android:id="@+id/switcherone"
        android:checked="false"
        android:layout_below="@+id/movetodown"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Score :"
        android:id="@+id/CUBESCORE"
        android:layout_below="@+id/switcherone"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="51dp"
        android:visibility="invisible" />

</RelativeLayout>
```

קובץ activity_cubertwo.xml

LEVEL 2

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.cubertwo"
    android:background="@drawable/back2">

    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="level 1? "
        android:id="@+id/switchertwo"
        android:checked="false"
        android:layout_below="@+id/AboveBallTwo"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:background="#ffffff"
        android:id="@+id/whiteTwo"
        android:visibility="visible"
        android:layout_marginLeft="65dp"
        android:layout_marginStart="65dp"
        android:layout_centerVertical="true"
        android:layout_toRightOf="@+id/switchertwo"
        android:layout_toEndOf="@+id/switchertwo" />

    <Button
        android:id="@+id/AboveBallOne"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:visibility="visible"
        android:layout_alignParentTop="true"
        android:layout_alignLeft="@+id/yellowOne"
        android:layout_alignStart="@+id/yellowOne"
        android:layout_marginRight="15dp"
        android:layout_marginLeft="15dp" />

    <Button
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:background="#ffffff"
        android:id="@+id/whiteOne"
        android:visibility="visible"
        android:layout_centerVertical="true"
        android:layout_alignLeft="@+id/yellowOne"
        android:layout_alignStart="@+id/yellowOne" />

    <Button
```

```
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:background="#ffff00"
    android:id="@+id/yellowOne"
    android:visibility="visible"
    android:layout_marginLeft="131dp"
    android:layout_marginStart="131dp"
    android:layout_above="@+id/whiteOne"
    android:layout_toRightOf="@+id/whiteTwo"
    android:layout_toEndOf="@+id/whiteTwo" />

<Button
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:background="#ffff00"
    android:id="@+id/yellowTwo"
    android:visibility="visible"
    android:layout_alignTop="@+id/yellowOne"
    android:layout_alignLeft="@+id/whiteTwo"
    android:layout_alignStart="@+id/whiteTwo" />

<Button
    android:id="@+id/AboveBallTwo"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:visibility="visible"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/yellowTwo"
    android:layout_alignEnd="@+id/yellowTwo"
    android:layout_marginRight="15dp" />

<Button
    android:id="@+id/DownBallOne"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:visibility="visible"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/AboveBallOne"
    android:layout_alignStart="@+id/AboveBallOne" />

<Button
    android:id="@+id/DownBallTwo"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:visibility="visible"
    android:layout_alignTop="@+id/DownBallOne"
    android:layout_alignLeft="@+id/AboveBallTwo"
    android:layout_alignStart="@+id/AboveBallTwo" />

</RelativeLayout>
```

קובץ activity_colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.colors"
    android:background="@drawable/mk">

    <Button
        android:layout_width="183dp"
        android:layout_height="300dp"
        android:id="@+id/btnRED"
        android:background="@drawable/red"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="false" />

    <Button
        android:layout_width="200dp"
        android:layout_height="300dp"
        android:id="@+id/btnGREEN"
        android:background="@drawable/green"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/btnRED"
        android:layout_toStartOf="@+id/btnRED" />

    <Button
        android:layout_width="200dp"
        android:layout_height="300dp"
        android:id="@+id/btnBLUE"
        android:background="@drawable/blue"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/btnGREEN"
        android:layout_toStartOf="@+id/btnGREEN" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Score :"
        android:id="@+id/t3"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:textSize="50dp"
        android:textColor="#ffffffff"
        android:background="@android:drawable/zoom_plate"
        android:textStyle="bold|normal" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/ourcolor"
        android:layout_below="@+id/t3"
        android:layout_alignLeft="@+id/t3"
```

```

        android:layout_alignStart="@+id/t3"
        android:visibility="invisible" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnagaining"
        android:background="@drawable/clienticon"
        android:layout_alignTop="@+id/ourcolor"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="200dp"
        android:layout_height="300dp"
        android:id="@+id	btnWHITE"
        android:background="@drawable/white"
        android:layout_above="@+id	btnRED"
        android:layout_toLeftOf="@+id	btnRED"
        android:layout_toStartOf="@+id	btnRED" />

    <Button
        android:layout_width="183dp"
        android:layout_height="300dp"
        android:id="@+id	btnBLACK"
        android:background="@drawable/black"
        android:layout_alignTop="@+id	btnWHITE"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_toRightOf="@+id	btnGREEN"
        android:layout_toEndOf="@+id	btnGREEN" />

    <Button
        android:layout_width="200dp"
        android:layout_height="300dp"
        android:id="@+id	btnYELLOW"
        android:background="@drawable/yellow"
        android:layout_above="@+id	btnBLUE"
        android:layout_toLeftOf="@+id	btnWHITE"
        android:layout_toStartOf="@+id	btnWHITE" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello , Thats me TAGU, i am here to help you,press me to listen
again."
        android:id="@+id/textView2"
        android:textSize="20dp"
        android:textColor="#ffffffff"
        android:background="@android:drawable/zoom_plate"
        android:layout_below="@+id/t3"
        android:layout_toRightOf="@+id/btnagaining"
        android:layout_toEndOf="@+id/btnagaining" />

    <Button
        android:layout_width="100dp"
        android:layout_height="70dp"
        android:id="@+id/skip"
        android:background="@drawable/taguwife"
        android:layout_alignTop="@+id/textView8"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello , Thats TAGU wife, i am here to help you,press me to SKIP
this round."
    android:id="@+id/textView8"
    android:textSize="20dp"
    android:textColor="#ffffffff"
    android:background="@android:drawable/zoom_plate"
    android:layout_below="@+id/btnagaining"
    android:layout_alignLeft="@+id/textView2"
    android:layout_alignStart="@+id/textView2" />

</RelativeLayout>
```

קובץ activity_clock.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.clock"
    android:id="@+id/r11"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/clockee" >

    <ImageButton
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/imageButton1"
        android:background="@drawable/t111"
        android:layout_marginLeft="53dp"
        android:layout_marginStart="53dp"
        android:layout_above="@+id/imageButton2"
        android:layout_toRightOf="@+id/imageButton12"
        android:layout_toEndOf="@+id/imageButton12"
        android:layout_marginBottom="46dp" />

    <ImageButton
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/imageButton12"
        android:contextClickable="false"
        android:layout_above="@+id/imageButton11"
        android:layout_centerHorizontal="true"
        android:background="@drawable/t111" />

    <ImageButton
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/imageButton11"
        android:background="@drawable/t111"
        android:layout_alignTop="@+id/imageButton1"
        android:layout_toLeftOf="@+id/imageButton12"
        android:layout_toStartOf="@+id/imageButton12" />

    <ImageButton
        android:layout_width="150dp"
        android:layout_height="70dp"
        android:id="@+id/imageButton2"
        android:background="@drawable/t111"
        android:layout_marginBottom="67dp"
        android:layout_above="@+id/imageButton3"
        android:layout_alignLeft="@+id/rstryagain"
        android:layout_alignStart="@+id/rstryagain" />

    <ImageButton
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:id="@+id/imageButton10"
        android:background="@drawable/t111"
        android:layout_below="@+id/imageButton1" />
```

```
    android:layout_toLeftOf="@+id/imageButton11"
    android:layout_toStartOf="@+id/imageButton11" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton3"
    android:background="@drawable/t111"
    android:layout_centerVertical="true"
    android:layout_toRightOf="@+id/imageButton1"
    android:layout_toEndOf="@+id/imageButton1" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton9"
    android:background="@drawable/t111"
    android:layout_alignTop="@+id/imageButton3"
    android:layout_toLeftOf="@+id/imageButton10"
    android:layout_toStartOf="@+id/imageButton10" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton4"
    android:background="@drawable/t111"
    android:layout_above="@+id/imageButton5"
    android:layout_alignLeft="@+id/imageButton3"
    android:layout_alignStart="@+id/imageButton3"
    android:layout_marginBottom="36dp" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton8"
    android:background="@drawable/t111"
    android:layout_alignTop="@+id/imageButton4"
    android:layout_toRightOf="@+id/imageButton9"
    android:layout_toEndOf="@+id/imageButton9" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton7"
    android:background="@drawable/t111"
    android:layout_above="@+id/imageButton6"
    android:layout_toRightOf="@+id/imageButton8"
    android:layout_toEndOf="@+id/imageButton8" />

<ImageButton
    android:layout_width="100dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton5"
    android:background="@drawable/t111"
    android:layout_above="@+id/imageButton6"
    android:layout_toRightOf="@+id/imageButton6"
    android:layout_toEndOf="@+id/imageButton6" />

<ImageButton
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:id="@+id/imageButton6"
    android:background="@drawable/t111"
```

```
        android:layout_above="@+id/rstryagain"
        android:layout_centerHorizontal="true" />

    <RadioButton
        android:id="@+id/radioButton44"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="false"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Reset And Try Again . . ."
        android:id="@+id/rstryagain"
        android:layout_marginBottom="45dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="#fbc4c4"/>

    <TextView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:id="@+id/txtClock"
        android:layout_below="@+id/imageButton6"
        android:layout_toLeftOf="@+id/imageButton7"
        android:layout_toStartOf="@+id/imageButton7"
        android:textSize="25dp" />

</RelativeLayout>
```

קובץ shaking.xml

קובץ עזר למשחק השעון במקרה של טעות המסך רועד

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <rotate
        android:duration="70"
        android:fromDegrees="-5"
        android:pivotX="50%"
        android:pivotY="50%"
        android:repeatCount="5"
        android:repeatMode="reverse"
        android:interpolator="@android:anim/linear_interpolator"
        android:toDegrees="5" />
</set>
```

קובץ activity_x.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.XO"
    android:background="@drawable/yes6">

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/aetsov3"
        android:layout_centerVertical="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/xxo" />

    <Button
        android:layout_width="40dp"
        android:layout_height="500dp"
        android:id="@+id/aetsov1"
        android:layout_marginLeft="137dp"
        android:layout_marginStart="137dp"
        android:background="@drawable/xxo"
        android:layout_alignTop="@+id/aetsov2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="40dp"
        android:layout_height="500dp"
        android:id="@+id/aetsov2"
        android:background="@drawable/xxo"
        android:layout_alignTop="@+id/b2"
        android:layout_toRightOf="@+id/b2"
        android:layout_toEndOf="@+id/b2" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:background="@drawable/xxo"
        android:layout_marginBottom="137dp"
        android:layout_alignBottom="@+id/aetsov1"
        android:layout_alignParentLeft="true" />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_above="@+id/aetsov1"
        android:id="@+id/linearLayout">
```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="choose X or O :"
    android:id="@+id/chooseXO"
    android:layout_gravity="center_horizontal"
    android:textSize="25dp"
    android:textColor="#000000"
    android:autoText="false" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/buttonX"
    android:background="@drawable/xxw" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_above="@+id/aetsov1"
    android:layout_alignParentTop="true">

    <Button
        android:layout_width="109dp"
        android:layout_height="117dp"
        android:id="@+id/buttonO"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="@drawable/o"
        android:layout_toEndOf="@+id/aetsov2" />
    </LinearLayout>
</LinearLayout>

<Button
    android:layout_width="140dp"
    android:layout_height="120dp"
    android:id="@+id/b1"
    android:layout_above="@+id/aetsov3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:background="@drawable/t111" />

<Button
    android:layout_width="170dp"
    android:layout_height="120dp"
    android:id="@+id/b2"
    android:layout_alignBottom="@+id/b1"
    android:layout_toRightOf="@+id/aetsov1"
    android:layout_toEndOf="@+id/aetsov1"
    android:background="@drawable/t111" />

<Button
    android:layout_width="180dp"
    android:layout_height="120dp"
    android:id="@+id/b3"
    android:layout_alignBottom="@+id/b2"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"

```

```
    android:background="@drawable/t111" />

<Button
    android:layout_width="140dp"
    android:layout_height="140dp"
    android:id="@+id/b4"
    android:layout_above="@+id/button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:background="@drawable/t111" />

<Button
    android:layout_width="170dp"
    android:layout_height="140dp"
    android:id="@+id/b5"
    android:layout_alignBottom="@+id/b4"
    android:layout_toRightOf="@+id/aetsov1"
    android:layout_toEndOf="@+id/aetsov1"
    android:background="@drawable/t111" />

<Button
    android:layout_width="180dp"
    android:layout_height="140dp"
    android:id="@+id/b6"
    android:layout_alignTop="@+id/b4"
    android:layout_alignRight="@+id/button"
    android:layout_alignEnd="@+id/button"
    android:background="@drawable/t111" />

<Button
    android:layout_width="170dp"
    android:layout_height="130dp"
    android:id="@+id/b8"
    android:layout_alignBottom="@+id/aetsov1"
    android:layout_toRightOf="@+id/aetsov1"
    android:layout_toEndOf="@+id/aetsov1"
    android:background="@drawable/t111" />

<Button
    android:layout_width="140dp"
    android:layout_height="130dp"
    android:id="@+id/b7"
    android:layout_alignBottom="@+id/aetsov1"
    android:layout_alignRight="@+id/b4"
    android:layout_alignEnd="@+id/b4"
    android:background="@drawable/t111" />

<Button
    android:layout_width="170dp"
    android:layout_height="130dp"
    android:id="@+id/b9"
    android:layout_alignTop="@+id/b7"
    android:layout_alignRight="@+id/button"
    android:layout_alignEnd="@+id/button"
    android:background="@drawable/t111" />
</RelativeLayout>
```

קובץ activity_store.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.Store"
    android:background="@drawable/ooo">

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:text="Typing Feature - Buy it!"
        android:id="@+id/btnTyping"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_marginTop="70dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/red" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:text="Photos Feature - Buy it!"
        android:id="@+id/btnPhotos"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_below="@+id/btnTyping"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/green" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:text="Math Feature - Buy it!"
        android:id="@+id/btnMathh"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_below="@+id/btnPhotos"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="@drawable/yellow" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:text="Clock Feature - BUY IT!"
        android:id="@+id/btnClock"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_below="@+id/btnMathh"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
```

```
    android:background="@drawable/white" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:text="Colors Feature - BUY IT!"
    android:id="@+id/btnColors"
    android:enabled="true"
    android:textSize="20dp"
    android:layout_below="@+id/btnClock"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:background="@drawable/blue" />

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="Feature"
    android:id="@+id/btnFeature"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textSize="28dp"
    android:background="@drawable/grey"
    android:layout_above="@+id/typeprice" />

<Button
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:text="SCORES"
    android:id="@+id/btnScores"
    android:textSize="30dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:background="@drawable/grey"
    android:layout_above="@+id/typescore" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:id="@+id/typescore"
    android:enabled="true"
    android:textSize="20dp"
    android:layout_alignTop="@+id/btnTyping"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:background="@drawable/red" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:id="@+id/photoscore"
    android:enabled="true"
    android:textSize="20dp"
    android:layout_below="@+id/typescore"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:background="@drawable/green" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
```

```
        android:id="@+id/mathscore"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_above="@+id/btnClock"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="@drawable/yellow" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:id="@+id/clockscore"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_above="@+id/btnColors"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="@drawable/white" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:id="@+id/colorscore"
        android:enabled="true"
        android:textSize="20dp"
        android:layout_below="@+id/clockscore"
        android:layout_alignLeft="@+id/clockscore"
        android:layout_alignStart="@+id/clockscore"
        android:background="@drawable/blue" />

    <Button
        android:layout_width="120dp"
        android:layout_height="70dp"
        android:text="PRICE"
        android:id="@+id/btnPrice"
        android:textSize="30dp"
        android:background="@drawable/grey"
        android:layout_alignParentTop="true"
        android:layout_alignRight="@+id/typeprice"
        android:layout_alignEnd="@+id/typeprice" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:id="@+id/typeprice"
        android:enabled="true"
        android:textSize="30dp"
        android:layout_above="@+id/btnPhotos"
        android:layout_centerHorizontal="true"
        android:text="20"
        android:background="@drawable/paper"
        android:layout_below="@+id/btnPrice" />

    <Button
        android:layout_width="120dp"
        android:layout_height="150dp"
        android:id="@+id/pricephotos"
        android:enabled="true"
        android:textSize="30dp"
        android:text="30"
        android:layout_below="@+id/typeprice"
        android:layout_centerHorizontal="true"
        android:background="@drawable/paper" />
```

```
<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:id="@+id/pricemath"
    android:enabled="true"
    android:textSize="30dp"
    android:text="12"
    android:layout_above="@+id/btnClock"
    android:layout_alignLeft="@+id/pricephotos"
    android:layout_alignStart="@+id/pricephotos"
    android:background="@drawable/paper" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:id="@+id/priceclock"
    android:enabled="true"
    android:textSize="30dp"
    android:text="28"
    android:layout_below="@+id/btnMathh"
    android:layout_centerHorizontal="true"
    android:background="@drawable/paper" />

<Button
    android:layout_width="120dp"
    android:layout_height="150dp"
    android:id="@+id/colorprice"
    android:enabled="true"
    android:textSize="30dp"
    android:text="120"
    android:layout_below="@+id/btnClock"
    android:layout_centerHorizontal="true"
    android:background="@drawable/paper" />
</RelativeLayout>
```

קובץ activity_user_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.UserInfo"
    android:background="@drawable/ttt">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="My Name : "
        android:id="@+id/MyName"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:textSize="40dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="My Scores : "
        android:id="@+id/MyScore"
        android:textSize="40dp"
        android:layout_below="@+id/MyName"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Change your password"
        android:id="@+id/ChangePass"
        android:layout_below="@+id/dtPassword"
        android:layout_alignRight="@+id/MyName"
        android:layout_alignEnd="@+id/MyName" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Email : "
        android:id="@+id/dtEmail"
        android:textSize="40dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/MyScore" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Password :"
```

```
    android:id="@+id/dtPassword"
    android:textSize="40dp"
    android:layout_below="@+id/dtEmail"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />
</RelativeLayout>
```

קובץ activity_about.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fares.taguapp.About"
    android:background="@drawable/grey">

    <VideoView
        android:layout_width="fill_parent"
        android:layout_height="350dp"
        android:id="@+id/videoView" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Enriching the kids experience of learning while playing and
enjoying their time - We Are TAGU!"
        android:id="@+id/textView7"
        android:textSize="30dp"
        android:textStyle="bold"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:textColor="#ab000000" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play!"
        android:id="@+id/playvideo"
        android:layout_above="@+id/textView7"
        android:layout_alignLeft="@+id/videoView"
        android:layout_alignStart="@+id/videoView"
        android:background="@drawable/colorbutton" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pause!"
        android:id="@+id/pausevideo"
        android:layout_above="@+id/textView7"
        android:layout_toRightOf="@+id/playvideo"
        android:layout_toEndOf="@+id/playvideo"
        android:background="@drawable/colorbutton" />
</RelativeLayout>
```

JAVA קבצי

קובץ welcome.java

```
package com.example.fares.taguapp;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

public class welcome extends AppCompatActivity {
    ImageView iv; // ImageView
    Animation animate,animate2; // Animation
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
        iv=(ImageView) findViewById(R.id.WelcomePhoto); // imageview
        animate = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.rotate); // first animation
        animate2 =
        AnimationUtils.loadAnimation(getApplicationContext(),R.anim.abc_fade_out); // second animation
        iv.startAnimation(animate); // running the animation
        animate.setAnimationListener(new Animation.AnimationListener() {
            @Override
            public void onAnimationStart(Animation animation) {
            }
            @Override
            public void onAnimationEnd(Animation animation) {
                iv.startAnimation(animate2); // running the animation
                finish(); // to end the Activity immediatly.
                Intent i = new Intent(getApplicationContext(), Signin.class); // new intent
                startActivity(i); // go to the sign in
            }
            @Override
            public void onAnimationRepeat(Animation animation) {
            }
        });
    }
}
```

קובץ Signin.java

```
package com.example.fares.taguapp;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.media.MediaPlayer;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.Random;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Signin extends AppCompatActivity {
    // we declare variables ... buttons and txtView and editText and helper VAR
    static EditText etName1,etPass1,etNewPass1,etNewName1,etNewMail; // edittext
    TextView name,pass,newname,newpass,checkingpassword,forgotpass; // textview
    Button SignIn,SignUp; // buttons
    static int j=0; // static one - use it with TAGUAPP Page.
    int indexForPassword,flag=0,flagletter=0,flagnumbers=0,i; // integer
    boolean key=false;
    // this array is for helping us to make a random 8 characters for password.
    String [] MakerPasswords = new
    String[] {"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T",
    "U","V","W","X","Y","Z"};
    Random rand = new Random(); // Random Numbers..
    String Usering="",Mailing="",MakeAnotherPassword="",s1,s2,str,str2,str3;
    @Override
    protected void onStop() {
        super.onStop();
        if(TAGUAPP.power==1)
            return;
        if(key==false)
            TAGUAPP.mysound.stop(); // turn OFF the music.
        else
            key=false;
    }
    @Override
    public void onBackPressed() {
        return; // you press the second button ( on tablet ) to exit the application.
    }
    // isEmailValid function - To Check if we REALLY have an email.
    public static boolean isEmailValid(String email) {
        boolean AreWeHave = false;
        String expression = "^\w\.-+@(\w\.-+\.)+[A-Z]{2,4}$"; // included ALL
CHARACTERS.
        CharSequence mailString = email; // a readable sequence of char values - we
can read CHAR values.
        // The resulting PATTERN can be used to create a MATCHER OBJECT that can match
arbitrary character sequences against the regular expression.
        Pattern pattern = Pattern.compile(expression, Pattern.CASE_INSENSITIVE); //
```

```

    // A MATCHER is created from a PATTERN by invoking the pattern's matcher
method.
    Matcher matcher = pattern.matcher(mailString);
    if (matcher.matches()) {
        AreWeHave = true;
    }
    return AreWeHave; // true/false.
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_signin);
    etNewMail = (EditText) findViewById(R.id.etylourEmail); // VAR Settings...
    etName1 = (EditText) findViewById(R.id.etName); // VAR Settings...
    etPass1 = (EditText) findViewById(R.id.etpassword); // VAR Settings...
    etNewName1 = (EditText) findViewById(R.id.etnewname); // VAR Settings...
    etNewPass1 = (EditText) findViewById(R.id.etnewpassword); // VAR Settings...
    SignIn = (Button) findViewById(R.id.btnSignIn); // VAR Settings...
    SignUp = (Button) findViewById(R.id.btnSignUp); // VAR Settings...
    forgotpass = (TextView) findViewById(R.id.txtforgoturpass); // VAR Settings...
    name = (TextView) findViewById(R.id.txtName); // VAR Settings...
    pass = (TextView) findViewById(R.id.txtPassword); // VAR Settings...
    newname = (TextView) findViewById(R.id.txtnewname); // VAR Settings...
    newpass = (TextView) findViewById(R.id.txtnewpassword); // VAR Settings...
    checkingpassword = (TextView) findViewById(R.id.checkpass); // VAR Settings...
    checkingpassword.setEnabled(false); // show you how strong your pass is.
    Callanimate(); // CALL Animation.
    if (TAGUAPP.key != 1) {
        TAGUAPP.mysound = MediaPlayer.create(this, R.raw.song); // we play
music.
        TAGUAPP.mysound.start();
        TAGUAPP.flag = false;
    }
    etName1.requestFocus(); // we AUTOMATICALLY put the FOCUS on the edittext of
Name.
    my_listener m1 = new my_listener(); // my_listener class - we need this
object to connect the variables to the work.
    SignIn.setOnClickListener(m1);
    SignUp.setOnClickListener(m1);
    forgotpass.setOnClickListener(m1);
    etNewPass1.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int
after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int
count) {
            checkingpassword.setEnabled(false);
        }
        // here we are checking the password according to the password RULES.
        /* 1. Minimum of 8 characters in length.
           2. Password is FAIR if the length = 8.
           3. Password is WEAK if Password LENGTH > 8 AND Password JUST includes
[a-z]/[A-Z] WITHOUT [0-9]
           4. Password is STRONG if Password LENGTH > 8 AND Password includes [a-
z]/[A-Z] AND [0-9]
        */
        @Override
        public void afterTextChanged(Editable s) {
            String Passing = etNewPass1.getText().toString();
            for (indexForPassword = 0; indexForPassword <= etNewPass1.length() -
1; indexForPassword++)

```

```

        {
            if((Passing.charAt(indexForPassword)>='a'
&&Passing.charAt(indexForPassword)<='z') || (Passing.charAt(indexForPassword)>='A' &&
Passing.charAt(indexForPassword)<='Z'))
                flagletter=1;
            if(Passing.charAt(indexForPassword)>='0' &&
Passing.charAt(indexForPassword)<='9')
                flagnumbers=1;
        }
        if (etNewPass1.length() < 8) {
            checkingpassword.setText("Minimum of 8 characters in length"); //
accor
            checkingpassword.setTextColor(Color.RED); // set color
            checkingpassword.setEnabled(true); // show the password status
text
        } else {
            if (etNewPass1.length() == 8) {
                checkingpassword.setEnabled(true);
                checkingpassword.setText("FAIR");
                checkingpassword.setTextColor(Color.BLUE); // set color
            }
            else if(etNewPass1.length() > 8 && flagnumbers==0 &&
flagletter==1)
            {
                flagletter=0;
                checkingpassword.setEnabled(true);
                checkingpassword.setText("WEAK");
                checkingpassword.setTextColor(Color.MAGENTA);
            }
            else if(etNewPass1.length() > 8 && flagnumbers==1 &&
flagletter==1)
            {
                flagletter=flagnumbers=0;
                checkingpassword.setEnabled(true);
                checkingpassword.setText("Strong");
                checkingpassword.setTextColor(Color.GREEN);
            }
        }
        flagletter=flagnumbers=0; // reset helper variables for the next time.
    }
});
}
*/
ChangerPassword function - to change password if you forgot your password.
1. We make a random password with 8 characters.
2. User must confirm his username AND email.
3. If user confirmed his mail and username then he can log in into the system.
4. In Info Page - the user can see his new password.
*/
public void ChangerPassword()
{
    for(i=1;i<=8;i++)
        MakeAnotherPassword=MakeAnotherPassword+MakerPasswords[rand.nextInt(26)];
    AlertDialog.Builder b = new AlertDialog.Builder(Signin.this);
    final AlertDialog.Builder b2 = new AlertDialog.Builder(Signin.this);
    b.setTitle("Please enter your username :");
    b2.setTitle("Please enter your email :");
    final EditText input = new EditText(Signin.this);
    final EditText input2 = new EditText(Signin.this);
    b.setView(input);
    b.setPositiveButton("your mail...", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int whichButton) {

```

```

        Usering = input.getText().toString();
        b2.setView(input2);
        b2.setPositiveButton("Send New Password", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog,
int whichButton) {
        Mailing = input2.getText().toString();
        my_db sendnewpass = new
my_db(getApplicationContext(), "JAMIL", null, 1);
        // we check if the MAIL AND USERNAME is
TRUE.
        if (isEmailValid(Mailing)==true &&
sendnewpass.searchforgotyourpassword(Usering,Mailing)==true)
        {

sendnewpass.updateCurrentPassword(Usering,MakeAnotherPassword); // new password is
updated.
        etName1.setText(Usering); // it looks
like normal sign in ...

etPass1.setText(MakeAnotherPassword); //it looks like normal sign in ...
        MakeAnotherPassword="";
        Intent i1 = new
Intent(getApplicationContext(), TAGUAPP.class);
        Toast.makeText(Signin.this,
"Welcome," + Usering, Toast.LENGTH_LONG).show(); // Hello Again.
        startActivity(i1); // active
welcome message.
    }
    else // if mail OR username is FALSE.
    {
        Usering = Mailing = "";
        Toast.makeText(getApplicationContext(), "Username/Email is Incorrect!",
Toast.LENGTH_SHORT).show();
        ChangerPassword(); // we have nothing
to do - just try again by Recursion - another round.
    }
}
);
        b2.setNegativeButton("CANCEL", null); // cancel if you do
want to exit the "forgot your password" dialog.
        b2.create().show();
    }
);
        b.setNegativeButton("CANCEL", null); // cancel if you do want to exit the
"forgot your password" dialog.
        b.create().show();
    }
    class my_listener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.txtforgoturpass:
                    ChangerPassword(); // run forgot your password function - we must
use the username and email.
                    break;
                case R.id.btnSignIn: // pressing sign in button...
                    key=true;
                    String s1 = etName1.getText().toString(); // get name and password

```

```

string..
String s2 = etPass1.getText().toString();
my_db md1 = new my_db(getApplicationContext(), "JAMIL", null, 1);
// we must use our DATABASE to check the name & pass.
if (md1.searchSignIn(s1, s2) == true) // searchSignIn(Name, Pass)
function returns true if we've the name & pass correctly.
{
    if(TAGUAPP.flag==true) // checking if we were at TAGUAPP Page
( if we log out ) - cuz we have to care about MUSIC if its ON/OFF.
        j=1;
    Intent i1 = new Intent(getApplicationContext(),
TAGUAPP.class);
    Toast.makeText(Signin.this, "Welcome," + s1,
Toast.LENGTH_LONG).show(); // welcome toast message
        startActivity(i1);
} else {
    Toast.makeText(Signin.this, "Username/Password is not
correct!", Toast.LENGTH_LONG).show(); // unfortunately :(
        etName1.setText(""); // reset all EDITTEXT's.
        etPass1.setText("");
        etNewMail.setText("");
}
break;
case R.id.btnSignUp: // pressing sign out button ...
String str = etNewName1.getText().toString(); // get NEW name and
NEW password and E-Mail string..
String str2 = etNewPass1.getText().toString();
String str3=etNewMail.getText().toString();
if (isValidEmail(str3)==false) // if we insert wrong expression to
E-Mail.
{
    //reset all edittext'S.. and try again.
    etNewName1.setText("");
    etNewPass1.setText("");
    etNewMail.setText("");
    str=str2=str3="";
    Toast.makeText(getApplicationContext(),"Email is
wrong!",Toast.LENGTH_LONG).show();
}
else if(!(str.length()>=4 && str.length()<=10)) // Name Length
MUST be between 4 - 10 CHARACTERS.
{
    // RESET if LEN < 4 or LEN > 10.
    etNewName1.setText("");
    etNewPass1.setText("");
    etNewMail.setText("");
    str=str2=str3="";
    Toast.makeText(getApplicationContext(),"Username MUST contain
more characters!",Toast.LENGTH_LONG).show();
}
else if(flag==1) // Username MUST contain JUST Numbers && Letters!
{
    // RESET...
    flag=0;
    etNewName1.setText("");
    etNewPass1.setText("");
    etNewMail.setText("");
    str=str2=str3="";
    Toast.makeText(getApplicationContext(),"Username MUST contain
JUST Numbers && Letters!",Toast.LENGTH_LONG).show();
}
else if(str2.length()<=3)
{

```

```

        // RESET...
        etNewName1.setText("");
        etNewPass1.setText("");
        etNewMail.setText("");
        str=str2=str3="";
        Toast.makeText(getApplicationContext(),"Password MUST contain more characters!",Toast.LENGTH_LONG).show();
    }
    else {
        /* if Name LENGTH between 4 - 10 characters
           AND PASS > 3 characters
           AND Email is correct
           THEN : We also check if name and email have not already exist.
               searchSignUp(NAME) function is checking if we do not already have this name.
               searchEmailSignUp(EMAIL) function is checking if we do not already have this email.
        */
        my_db md2 = new my_db(getApplicationContext(), "JAMIL", null, 1);
        if (md2.searchSignUp(str) == false &&
        md2.searchEmailSignUp(str3)==false)
        {
            md2.add_player(str, str2,str3); // add player to the PLAYER Table - and set score zero to math game.
            etNewName1.setText("");
            etNewPass1.setText("");
            etNewMail.setText("");
            Toast.makeText(Signin.this, "Username is created :)", Toast.LENGTH_LONG).show(); // New User.
        } else {
            // RESET ...
            etNewName1.setText("");
            etNewPass1.setText("");
            etNewMail.setText("");
            Toast.makeText(Signin.this, "Username/Email is already exists!", Toast.LENGTH_LONG).show(); // Sadness Message.
        }
    }
    break;
}
}
// Callanimate function - To Start Animations -> specific animation for specific objects.
public void Callanimate()
{
    SignUp.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    etPass1.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    etNewName1.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    etNewPass1.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    SignIn.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    etName1.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    name.startAnimation(AnimationUtils.loadAnimation(Signin.this,
    android.R.anim.slide_in_left));
    pass.startAnimation(AnimationUtils.loadAnimation(Signin.this,

```

```
        android.R.anim.slide_in_left));
        newname.startAnimation(AnimationUtils.loadAnimation(Signin.this,
        android.R.anim.slide_in_left));
        newpass.startAnimation(AnimationUtils.loadAnimation(Signin.this,
        android.R.anim.slide_in_left));
    }
}
```

קובץ TAGUAPP.java

```
package com.example.fares.taguapp;

import android.app.AlertDialog;
import android.content.ClipData;
import android.content.DialogInterface;
import android.content.Intent;
import android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.DragEvent;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class TAGUAPP extends AppCompatActivity {

    final int[] songs={R.raw.song,R.raw.song2,R.raw.song3};
    int SongIndexer=songs[0],top3,bot3,left3,right3,top4,bot4,left4,right4;// it's
his default for exceptions.
    Button
dragged,target,mathbtn,letterbtn,typingbtn,aboutbtn,signoutbtn,colors,btncontactus,foc
us,photos,clocking,URFace,XO,Info,Store; // buttons
    static Button sounding;
    static MediaPlayer mysound; //
    static boolean flag;
    static int key=0,countersong=0,helpers,power=0; // we use key variable at the sign
in when we log out
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_taguapp);
        key=0;
        if(countersong!=0)
            SongIndexer=helpers;
        mathbtn=(Button)findViewById(R.id.math); // setting find view by id
        Info=(Button)findViewById(R.id.userinfo); // setting find view by id
        Store=(Button)findViewById(R.id.btnStore); // setting find view by id
        clocking=(Button)findViewById(R.id.centers); // setting find view by id
        btncontactus=(Button)findViewById(R.id.contactus); // setting find view by id
        letterbtn=(Button)findViewById(R.id.letter); // setting find view by id
        colors=(Button)findViewById(R.id.btnco); // setting find view by id
        typingbtn=(Button)findViewById(R.id.typing); // setting find view by id
        XO=(Button)findViewById(R.id.btnXO); // setting find view by id
        URFace=(Button)findViewById(R.id.cubing); // setting find view by id
        aboutbtn=(Button)findViewById(R.id.about); // setting find view by id
        signoutbtn=(Button)findViewById(R.id.signout); // setting find view by id
        sounding=(Button)findViewById(R.id.sound); // setting find view by id
        photos=(Button)findViewById(R.id.photos1); // setting find view by id
        focus=(Button)findViewById(R.id.focus1); // setting find view by id
        if(flag==true) {
            sounding.setBackgroundResource(R.drawable.stopmus); // changing icon ...
            mysound.pause(); // stop music ...
        }
        my_listener m1 = new my_listener(); // we run the my_listener class ...
        Store.setOnClickListener(m1); // connect buttons to the my_listener class
        mathbtn.setOnClickListener(m1);
        Info.setOnClickListener(m1);
        colors.setOnClickListener(m1);
        letterbtn.setOnClickListener(m1);
    }
}
```

```

typingbtn.setOnClickListener(m1);
aboutbtn.setOnClickListener(m1);
signoutbtn.setOnClickListener(m1);
btncontactus.setOnClickListener(m1);
focus.setOnClickListener(m1);
photos.setOnClickListener(m1);
clocking.setOnClickListener(m1);
URFace.setOnClickListener(m1);
XO.setOnClickListener(m1);

longerclicking kk = new longerclicking(); // JUST FOR SOUND
sounding.setOnLongClickListener(kk); // connect button to the longerclicking
class

    LongPressListener m2 = new LongPressListener(); // we run the LongPressListener
class ...
    mathbtn.setOnLongClickListener(m2); // connect buttons to the
LongPressListener class
    letterbtn.setOnLongClickListener(m2);
    colors.setOnLongClickListener(m2);
    focus.setOnLongClickListener(m2);
    photos.setOnLongClickListener(m2);
    clocking.setOnLongClickListener(m2);
    typingbtn.setOnLongClickListener(m2);
    URFace.setOnLongClickListener(m2);
    XO.setOnLongClickListener(m2);
    signoutbtn.setOnLongClickListener(m2);
    aboutbtn.setOnLongClickListener(m2);
    Store.setOnLongClickListener(m2);
    btncontactus.setOnLongClickListener(m2);
    Info.setOnLongClickListener(m2);

    DragListener d = new DragListener(); // we run the DragListener class ...
    mathbtn.setOnDragListener(d); // connect buttons to the DragListener class
    btncontactus.setOnDragListener(d);
    Store.setOnDragListener(d);
    letterbtn.setOnDragListener(d);
    colors.setOnDragListener(d);
    focus.setOnDragListener(d);
    Info.setOnDragListener(d);
    photos.setOnDragListener(d);
    clocking.setOnDragListener(d);
    typingbtn.setOnDragListener(d);
    URFace.setOnDragListener(d);
    XO.setOnDragListener(d);
    signoutbtn.setOnDragListener(d);
    aboutbtn.setOnDragListener(d);
}
@Override
public void onBackPressed() {
    // super.onBackPressed();
    // Do Nothing - we freeze the back pressed button.
    return;
}
@Override
protected void onStop() {
    super.onStop();
    key=1;
    countersong++;
    helpers=SongIndexer; // we save the current song and if we log out and back to
Sign in Page the current song is still playing ...
}
class longerclicking implements View.OnLongClickListener{

```

```

@Override
public boolean onLongClick(View v) {
    switch (v.getId()) {
        case R.id.sound: // long click to show music list ...
            final String[] items = {"Song1", "Song2", "Song3"}; // songs array
            AlertDialog.Builder builder = new
AlertDialog.Builder(TAGUAPP.this); // alert dialog to choose song.
            builder.setTitle("Choose your song");
            builder.setItems(items, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int item) {
                    if(items[0]==items[item]) {
                        mysound.pause(); // stoppnig ...
                        mysound = MediaPlayer.create(TAGUAPP.this, songs[0]);
// create and set song
                        SongIndexer=songs[0]; // we save the current song -
because we want keep it playing when we log out
                        mysound.start(); // running ...
                        Toast.makeText(getApplicationContext(), "Song1",
Toast.LENGTH_LONG).show();
                    }
                    else if(items[1]==items[item]) {
                        mysound.pause(); // stoppnig ...
                        mysound = MediaPlayer.create(TAGUAPP.this,
songs[1]); // create and set song
                        SongIndexer=songs[1]; // we save the current song -
because we want keep it playing when we log out
                        mysound.start(); // running ...
                        Toast.makeText(getApplicationContext(), "Song2",
Toast.LENGTH_LONG).show();
                    }
                    else if(items[2]==items[item]) {
                        mysound.pause(); // stoppnig ...
                        mysound = MediaPlayer.create(TAGUAPP.this,
songs[2]); // create and set song
                        SongIndexer=songs[2]; // we save the current song -
because we want keep it playing when we log out
                        mysound.start(); // running ...
                        Toast.makeText(getApplicationContext(), "Song3",
Toast.LENGTH_LONG).show();
                    }
                }
            });
            AlertDialog alert = builder.create(); // Alert Dialog
            alert.show();
            sounding.setBackgroundResource(R.drawable.startmusic); // changing
icon ...
            break;
        }
        return true;
    }
}
class my_listener implements View.OnClickListener {
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnStore://GO TO STORE
Intent Storetransfer=new
Intent(getApplicationContext(),Store.class) ;
startActivity(Storetransfer);
break;
        case R.id userinfo://GO TO INFO
Intent Infotransfer=new

```

```

Intent(getApplicationContext(),UserInfo.class) ;
        startActivity(Infotransfer);
        break;
    case R.id.btnXO://GO TO X&O
        Intent XOTransfer=new Intent(getApplicationContext(),XO.class) ;
        startActivity(XOTransfer);
        break;
    case R.id.photos1://GO TO PHOTOS
        Intent phototransfer=new
Intent(getApplicationContext(),photoss.class) ;
        startActivity(phototransfer);
        break;
    case R.id.focus1://GO TO FOCUS
        Intent focustransfer=new Intent(getApplicationContext(),
com.example.fares.taguapp.focus.class) ;
        startActivity(focustransfer);
        break;
    case R.id.centers://GO TO CLOCK
        Intent clockingtransfer=new
Intent(getApplicationContext(),clock.class) ;
        startActivity(clockingtransfer);
        break;
    case R.id.math://GO TO MATH
        Intent mathtransfer=new Intent(getApplicationContext(),Math.class)
;

        startActivity(mathtransfer);
        break;
    case R.id.btnco://GO TO COLORS
        Intent colorTransfer=new
Intent(getApplicationContext(),colors.class) ;
        startActivity(colorTransfer);
        break;
    case R.id.signout: // log out -> reseting all flags (for the next
user) and saving the current music and go back to sign in page.
        com.example.fares.taguapp.Store.flagPhotos=
            com.example.fares.taguapp.Store.flagClock=
                com.example.fares.taguapp.Store.flagColors=
                    com.example.fares.taguapp.Store.flagMath=


        com.example.fares.taguapp.Store.flagType=0;
            key=1;
            countersong++;
            helpers=SongIndexer; // saving the current song
            power=1;
            Intent back=new Intent(getApplicationContext(),Signin.class);
            startActivity(back);
            break;
    case R.id.letter:
        Intent lettertransfer=new
Intent(getApplicationContext(),letters.class);
        startActivity(lettertransfer);
        break;
    case R.id.typing:
        Intent typingtransfer=new
Intent(getApplicationContext(),typing.class);
        startActivity(typingtransfer);
        break;
    case R.id.about:
        Intent abouttransfer=new
Intent(getApplicationContext(),About.class);
        startActivity(abouttransfer);
        break;
    case R.id.contactus: // here we use action send of intent to send an

```

```

email to ADMIN emails.
        Intent emailintent=new Intent(Intent.ACTION_SEND);
        emailintent.putExtra(Intent.EXTRA_EMAIL, new
String[]{"mr.nakhle@gmail.com"});
        emailintent.putExtra(Intent.EXTRA_CC, new
String[]{"fares.salameh@outlook.com"});
        emailintent.putExtra(Intent.EXTRA_SUBJECT,""); // subject text
        emailintent.putExtra(Intent.EXTRA_TEXT,"");// body message
        emailintent.setType("Message/rfc822"); // the entire message with
no alterations, and no headers missed, all the content intact.
        startActivityForResult(Intent.createChooser(emailintent, "choose email
client...")); // choose how u wanna send the message (GMAIL etc..)
        break;
    case R.id.cubing://GO TO CUBE GAME ...
        Intent IronDometransfer=new
Intent(getApplicationContext(),cubeIt.class);
        startActivityForResult(IronDometransfer); // going to cubeIt game.
        break;
    }
}
public void playMusic(View view) // playMusic function - to play/stop music.
{
    if(flag==false) {
        flag=true;
        sounding.setBackgroundResource(R.drawable.stopmus); // changing icon
        mysound.pause(); // stopping ...
    }
    else
    {
        flag = false;
        mysound.pause(); // pause it.
        mysound = MediaPlayer.create(TAGUAPP.this, SongIndexer);
        mysound.start(); // starting ...
        sounding.setBackgroundResource(R.drawable.startmusic); // changing icon
    }
}
//*****
class LongPressListener implements View.OnLongClickListener {
    @Override
    public boolean onLongClick(View view) {
        final ClipData data = ClipData.newPlainText("", ""); // Long Click to
catch the button ...
        View.DragShadowBuilder shadowBuilder = new View.DragShadowBuilder(view);
        view.startDrag(data, shadowBuilder, view ,0);
        view.setVisibility(View.VISIBLE); // show the dragging ...
        return true;
    }
}
class DragListener implements View.OnDragListener {
    @Override
    public boolean onDrag(View v, DragEvent event)
    {
        // we have two Buttons Variables which help us to swapping two buttons
places.
        // SWAPPING - we swap top,bottom,left,right of the two buttons
        target = (Button)v; // target button ...
        dragged=(Button)event.getLocalState(); // the current dragged button ...
        switch (event.getAction()) {
            case DragEvent.ACTION_DRAG_STARTED: // catch and save
top/bottom/left/right current button ...
                top3=dragged.getTop(); // top
                bot3=dragged.getBottom(); // bottom

```

```

        left3=dragged.getLeft(); // left
        right3=dragged.getRight(); // right
        break;
    case DragEvent.ACTION_DROP: // drop the button which we catched and
swap it with the button which we stand on.
        // swapping top/bottom/right/left for two buttons ...
        // we have 8 variables , to save the (
top3/4,bottom3/4,right3/4,left3/4 ) of the buttons.
        top4 = target.getTop();
        bot4 = target.getBottom();
        left4 = target.getLeft();
        right4 = target.getRight();
        target.setTop(top3);
        target.setBottom(bot3);
        target.setLeft(left3);
        target.setRight(right3);
        dragged.setTop(top4);
        dragged.setBottom(bot4);
        dragged.setLeft(left4);
        dragged.setRight(right4);
        break;
    case DragEvent.ACTION_DRAG_ENDED:
        break;
    }
    return true;
}
}
}

```

קובץ Math.java

```
package com.example.fares.taguapp;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextWatcher;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.AlphaAnimation;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.util.concurrent.TimeUnit;
public class Math extends AppCompatActivity {
    public Button ShowAgain, num1, num2, num3, num4, num5, num6, CLEAR; // buttons
    public TextView res, ResultShow, Formula, ChangeTheRange; // textView
    static TextView score; // textView - works with Store Page.
    int HivHoovFeature, j = 0, HivHoovFeature2, key = 0, NumberRange=15; // integers
    static int CharactercounterInFormula = 0, IndexerForNumbers = 0; // integers too..
    public int scores = 0;
    static boolean SwitchClickOneNumber = false, SwitchClickTwoNumber = false;
    public static int[] TwoNumInFormula = new int[2]; // two numbers
    static char c; // plus or minus
    String[] str = new String[3];
    String BuildingFormulaStr = "", stringres = "", NameString, PasswordString;
    final CounterClass timerMath = new CounterClass(2000, 1000); // timer for 2
    seconds
    final CounterHelper timerFeatures = new CounterHelper(15000, 1000); // timer for
    1.5 seconds
    final CounterCurrentAnswer timercurrentanswer = new CounterCurrentAnswer(3000,
    1000); // timer for 3 seconds
    int[] FormulaNumArray = new int[5];
    Animation SlideDownOut;
    MathClass mathing = new MathClass();
    @Override
    protected void onStop() {
        super.onStop();
        timerFeatures.cancel(); // stop timers ...
        timerMath.cancel(); // stop checking the answer
        timercurrentanswer.cancel(); // stop current answer
    }
    public void ChangerRange()
    {
        AlertDialog.Builder b = new AlertDialog.Builder(Math.this);
        b.setTitle("Please enter a range of numbers");
        final EditText input = new EditText(Math.this);
        b.setView(input);
        b.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int whichButton) {
                // SHOULD BE AN INTEGER VALUE AND BETWEEN 1 AND 100.
                try {
                    NumberRange=Integer.parseInt(input.getText().toString());
                }
            }
        });
    }
}
```

```

        if(NumberRange<=1 || NumberRange>100)
            ChangerRange(); // Recursion - do it again while the number is
not between the range.
        } catch (NumberFormatException e) {
            input.setText("");
            ChangerRange(); // Recursion ...
        }
    }
});  

b.setNegativeButton("CANCEL", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int whichButton) {
        ChangerRange();
    }
});
b.create().show();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_math);
    final my_db md66 = new my_db(getApplicationContext(), "JAMIL", null, 1); //  

calling DB ...
    ChangerRange(); // call ChangerRange function ...

SlideDownOut=AnimationUtils.loadAnimation(getApplicationContext(),R.anim.slide_down_out
t); // Slide Down Out Animation
    NameString = Signin.etName1.getText().toString(); // get current user name
    PasswordString = Signin.etPass1.getText().toString(); // get current password
    scores = md66.getscoremath(NameString, PasswordString); // get last update
score
    num1 = (Button) findViewById(R.id.bbb1); // button declaration ...
    num2 = (Button) findViewById(R.id.b22); // button declaration ...
    num3 = (Button) findViewById(R.id.b3); // button declaration ...
    num4 = (Button) findViewById(R.id.b4); // button declaration ...
    num5 = (Button) findViewById(R.id.b5); // button declaration ...
    num6 = (Button) findViewById(R.id.b6); // button declaration ...
    ShowAgain = (Button) findViewById(R.id.btnagain); // button declaration ...
    CLEAR = (Button) findViewById(R.id.Clear); // button declaration ...
    ChangeTheRange=(TextView) findViewById(R.id.txtchangerange); // txtView
declaration ...
    res = (TextView) findViewById(R.id.txtresult); // txtView declaration ...
    ResultShow = (TextView) findViewById(R.id.txtres); // txtView declaration ...
    score = (TextView) findViewById(R.id.txtScore); // txtView declaration ...
    Formula = (TextView) findViewById(R.id.txtnos7a2); // txtView declaration ...
    if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())>=1)
        ShowAgain.setEnabled(true); // show first feature if we have it.
    else
        ShowAgain.setEnabled(false); // do not show.
    if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
        timerFeatures.start(); // start helper timer if we have the second
feature.
        // timerFeatures - if we have second feature so we run this TIMER to check
when timer is DONE which button we have to Flickering.
        score.setText("Score : " + scores); // score text
        mathing.CreaterNumbers(FormulaNumArray, NumberRange); // create numbers by
CreaterNumbers
        mathing.ShowTheDetails(res, num1, num2, num3, num4, num5, num6,
FormulaNumArray); // put all the values on the buttons after CREATING NUMBERS.
    ChangeTheRange.setOnClickListener(new View.OnClickListener() {
        @Override

```

```

        public void onClick(View v) {
            ChangerRange(); // calling ChangerRange function.
        }
    });
ShowAgain.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timercurrentanswer.start();
        SwitchClickOneNumber = SwitchClickTwoNumber = false;
        stringres = ""; // reset ...
        ResultShow.setText(""); // reset ...
        if (CharactercounterInFormula == 3 && IndexerForNumbers == 2)
        { // reset all things because we chose to see the answer ...
            str[0] = str[1] = str[2] = BuildingFormulaStr = "";
            CharactercounterInFormula = IndexerForNumbers = TwoNumInFormula[0]
= TwoNumInFormula[1] = 0;
            c = ' ' ; // rest +-
        }
        Formula.setText(mathing.ShowResults(FormulaNumArray)); // show
possibilities answers ...

Formula.startAnimation(AnimationUtils.loadAnimation(getApplicationContext(),
android.R.anim.slide_in_left)); // show it animationally.
        key = 0;
        locks(false); // when application show us the possibilities answers - we
cant click any button.
        timerFeatures.cancel(); // timer sec feature STOP ...
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the second
feature.
        // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
    }
});
CLEAR.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        // we - here - CLEAR EVERYTHING.
        BuildingFormulaStr = "";
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // turn on again
the PLUS AND MINUS buttons.
        mathing.zero(TwoNumInFormula, str); // clear and reset
        c = ' ' ;
        Formula.setText("");
        CharactercounterInFormula = IndexerForNumbers = key = 0;
        timerFeatures.cancel(); // timer sec feature STOP ...
        if (md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 2)
            timerFeatures.start(); // start helper timer if we have the second
feature.
        // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
        locks(true); // turn ON.
    }
});
num1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timerFeatures.cancel();
        j = 0;
        stringres = "";
    }
});

```

```

        key++;
        ResultShow.setText(""); // reset ...
        if (IndexerForNumbers != 2) {
            TwoNumInFormula[IndexerForNumbers] =
Integer.parseInt(num1.getText().toString());
            str[CharactercounterInFormula] = num1.getText().toString(); // we
save this button value.
            BuildingFormulaStr += num1.getText().toString(); // shershoor for
the array.
            Formula.startAnimation(SlideDownOut); // animate it while adding
new character to FORMULA.
            Formula.setText(BuildingFormulaStr); //add new character
            CharactercounterInFormula++; // how many character in formula -
MAX is 3.
            IndexerForNumbers++; // to next index ...
        }
        if (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula) != 0) // running Checking Function
        {
            timerMath.start(); // we wait 2 seconds then the app tells us if we
are right or wrong.
            locks(false); // when application checks our answer - we cant click
any button.
        }
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the second
feature.
            // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
        }
    });
num2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timerFeatures.cancel();
        j = 0;
        stringres = "";
        key++;
        ResultShow.setText("");
        if (IndexerForNumbers != 2) {
            TwoNumInFormula[IndexerForNumbers] =
Integer.parseInt(num2.getText().toString());
            str[CharactercounterInFormula] = num2.getText().toString();
            BuildingFormulaStr += num2.getText().toString(); // shershoor for
the array.
            Formula.startAnimation(SlideDownOut); // animate it while adding
new character to FORMULA.
            Formula.setText(BuildingFormulaStr); //add new character
            CharactercounterInFormula++; // how many character in formula -
MAX is 3.
            IndexerForNumbers++;
        }
        if (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula) != 0) // running Checking Function
        {
            timerMath.start(); // we wait 2 seconds then the app tells us if we
are right or wrong.
            locks(true); // when application checks our answer - we cant click
any button.
        }
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)

```

```

        timerFeatures.start(); // start helper timer if we have the second
feature.
        // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
    }
})
num3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timerFeatures.cancel();
        j = 0;
        stringres = "";
        key++;
        ResultShow.setText("");
        if (IndexerForNumbers != 2) {
            str[CharactercounterInFormula] = num3.getText().toString();
            TwoNumInFormula[IndexerForNumbers] =
Integer.parseInt(num3.getText().toString());
            BuildingFormulaStr += num3.getText().toString(); // shershoor for
the array.
            Formula.startAnimation(SlideDownOut); // animate it while adding
new character to FORMULA.
            Formula.setText(BuildingFormulaStr); //add new character
            CharactercounterInFormula++; // how many character in formula - MAX
is 3.
            IndexerForNumbers++; // to next index ...
        }
        if (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula) != 0) // running Checking Function
        {
            timerMath.start(); // we wait 2 seconds then the app tells us if we
are right or wrong.
            locks(false); // when application checks our answer - we cant click
any button.
        }
        if (md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the second
feature.
            // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
        }
})
num4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        timerFeatures.cancel();
        j = 0;
        stringres = "";
        key++;
        ResultShow.setText("");
        if (IndexerForNumbers != 2) {
            str[CharactercounterInFormula] = num4.getText().toString();
            TwoNumInFormula[IndexerForNumbers] =
Integer.parseInt(num4.getText().toString());
            BuildingFormulaStr += num4.getText().toString(); // shershoor
for the array.
            Formula.startAnimation(SlideDownOut); // animate it while
adding new character to FORMULA.
            Formula.setText(BuildingFormulaStr); //add new character
            CharactercounterInFormula++; // how many character in formula -
MAX is 3.
            IndexerForNumbers++; // to next index ...

```

```

        }
        if (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula) != 0) // running Checking Function
        {
            timerMath.start(); // we wait 2 seconds then the app tells us if we
are right or wrong.
            locks(false); // when application checks our answer - we cant click
any button.
        }
        if (md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 2)
            timerFeatures.start(); // start helper timer if we have the
second feature.
            // timerFeatures - if we have second feature so we run this TIMER to
check when timer is DONE which button we have to Flickering.
        }
    });
num5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (SwitchClickOneNumber == false && BuildingFormulaStr.length() != 0 &&
CharactercounterInFormula!=2) {
            SwitchClickOneNumber = true;
            if (c != '-') {
                j = 1;
                c = '+';
                str[CharactercounterInFormula] = num5.getText().toString();
                BuildingFormulaStr += c; // shershoor for the array.
                Formula.startAnimation(SlideDownOut); // animate it while
adding new character to FORMULA.
                Formula.setText(BuildingFormulaStr); //add new character
                CharactercounterInFormula++; // how many character in formula -
MAX is 3.
                HivHoovFeature=mathing.ShowTheThirdCharacter(FormulaNumArray);
                timerFeatures.cancel();
                if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
                    timerFeatures.start(); // start helper timer if we have the
second feature.
                    // timerFeatures - if we have second feature so we run this
TIMER to check when timer is DONE which button we have to Flickering.
            } else {
                BuildingFormulaStr = ""; // reset formula...
                CharactercounterInFormula = IndexerForNumbers = j = 0;
                mathing.zero(TwoNumInFormula, str); // set ZERO to the array
that save the numbers.
                c = '+'; // reset +-
                Formula.setText(""); // reset formula edittext
                Toast.makeText(getApplicationContext(), "Number One +/- Number
Two", Toast.LENGTH_SHORT).show(); // error message.
                CharactercounterInFormula = IndexerForNumbers = key = 0; //
reset
                timerFeatures.cancel(); // STOP ...
                if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
                    timerFeatures.start(); // start helper timer if we have the
second feature.
                    // timerFeatures - if we have second feature so we run this
TIMER to check when timer is DONE which button we have to Flickering.
                    SwitchClickOneNumber = SwitchClickTwoNumber = false; // reset
to next step
                    CharactercounterInFormula = IndexerForNumbers = key = 0; //
reset
            }
        }
    }
});

```

```

        timerFeatures.cancel();
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the
second feature.
            // timerFeatures - if we have second feature so we run this
TIMER to check when timer is DONE which button we have to Flickering.
        }
    }
}
);
num6.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (SwitchClickTwoNumber == false && BuildingFormulaStr.length() !=0 &&
CharactercounterInFormula!=2) {
    SwitchClickTwoNumber = true;
    if (c != '+') {
        j = 1;
        c = '-'; // set minus
        str[CharactercounterInFormula] = num6.getText().toString();
        BuildingFormulaStr += c; // shershoor for the array.
        Formula.startAnimation(SlideDownOut); // animate it while
adding new character to FORMULA.
        Formula.setText(BuildingFormulaStr); //add new character
        CharactercounterInFormula++; // how many character in formula -
MAX is 3.
        HivHoovFeature=mathing.ShowTheThirdCharacter(FormulaNumArray);
        timerFeatures.cancel();
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the
second feature.
            // timerFeatures - if we have second feature so we run this
TIMER to check when timer is DONE which button we have to Flickering.
    } else {
        BuildingFormulaStr = "";
        CharactercounterInFormula = IndexerForNumbers = j = 0;
        mathing.zero(TwoNumInFormula, str); // set ZERO to the array
that save the numbers.
        c = '+'; // reset +/-
        Formula.setText(""); // reset formula edittext
        Toast.makeText(getApplicationContext(), "Number One +/- Number
Two", Toast.LENGTH_SHORT).show(); // error message ...
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // reset
to next step
        CharactercounterInFormula = IndexerForNumbers = key = 0; //
reset
        timerFeatures.cancel(); // STOP ...
        if(md66.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
            timerFeatures.start(); // start helper timer if we have the
second feature.
            // timerFeatures - if we have second feature so we run this
TIMER to check when timer is DONE which button we have to Flickering.
    }
}
}
);
}
public class CounterHelper extends CountDownTimer {
/**
 * @param millisInFuture      The number of millis in the future from the call

```

```

        *
        * to {@link #start()} until the countdown is done
and {@link #onFinish()}                                is called.
        *
        * @param countDownInterval The interval along the way to receive
        *                               {@link #onTick(long)} callbacks.
        */
    public CounterHelper(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    @Override
    public void onTick(long millisUntilFinished) {

    }
    // onFinish function - checking which character now is ready to Flickering (
if numberOne OR PLUS/MINUS OR numberTwo ) then run the Flickering.
    @Override
    public void onFinish()
    {
        if (BuildingFormulaStr.length()==0) // if we have no formula then run
Flickering on NUMBER ONE.
        {
            HivHoovFeature2=mathing.ShowTheFirstCharacter(FormulaNumArray);
            mathing.checkingthefirst(HivHoovFeature2, num1, num2, num3, num4);
        }
        else if(BuildingFormulaStr.length()==1 || (HivHoovFeature2>=10 &&
HivHoovFeature2<=99 && BuildingFormulaStr.length()==2) )
        { /* if we have just one character then run Flickering on NUMBER ONE.
           this one character is numberOne - numberOne could be ONE DIGITS OR TWO
DIGITS if the range of numbers is greater than 10.*/
            char ch=mathing.ShowTheSecondCharacter(FormulaNumArray);
            mathing.checkingthesecond(ch,num5,num6); // running Flickering.
        }
        else if( (HivHoovFeature>=10 && HivHoovFeature<=99 &&
BuildingFormulaStr.length()==3) || (BuildingFormulaStr.length()==2))
        {
            // if we have Formula Length is 3 and number one is two digits OR
TWO characters ( number one is one digit & plus or minus )
            mathing.checkingthefirst(HivHoovFeature,num1,num2,num3,num4);
        }
        timerFeatures.start(); // start helper timer if we have the second feature.
        // timerFeatures - if we have second feature so we run this TIMER to check
when timer is DONE which button we have to Flickering.
    }
}
public class CounterClass extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                           to {@link #start()} until the countdown is done
and {@link #onFinish()}                                is called.
     * @param countDownInterval The interval along the way to receive
     *                           {@link #onTick(long)} callbacks.
     */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {
        long millis = millisUntilFinished;
        String hms = String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis), TimeUnit.MILLISECONDS.toMinutes(millis) -
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),

```

```

        TimeUnit.MILLISECONDS.toSeconds(millis) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        // how the timer looks like ...
    }
    // this is the function which run when the time is done !
@Override
public void onFinish() {
    my_db md30 = new my_db(getApplicationContext(), "JAMIL", null, 1);
    timerFeatures.cancel(); // STOP ...
    switch (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula))
    {
        case 1: // checking which situation we have now ...
        case 3: // checking which situation we have now ...
        case 4: // checking which situation we have now ...
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // turn ON
plus/minus button ...
        scores++; // correct answer
        Toast.makeText(getApplicationContext(), "Correct !!!",
Toast.LENGTH_SHORT).show(); // great message
        // here - the application is telling you what features you have
and suggesting you to go to the store and buy it.
        if (scores >= Store.updatemath &&
md30.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 0)
            Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the Hint Feature!", Toast.LENGTH_SHORT).show();
        else if (scores >= Store.updatemath &&
md30.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 1)
            Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the Flircking Feature!", Toast.LENGTH_SHORT).show();
        my_db md3 = new my_db(getApplicationContext(), "JAMIL", null, 1);
// CALLING DB ...
        md3.updaterScoreMath(Signin.etName1.getText().toString(), scores);
// updating score
        break;
        case 2:
        case 5:
        timercurrentanswer.start(); // wrong answer - then the APP shows
you the correct answers!
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // turn ON
plus/minus button ...
        ShowAgain.setEnabled(false);
        Formula.setText(""); // reset
        Toast.makeText(getApplicationContext(), "Incorrect !!!",
Toast.LENGTH_SHORT).show(); // wrong message ...
        ResultShow.setText("result = " + FormulaNumArray[4] + "\n" +
mathing.ShowResults(FormulaNumArray)); // showing possibilities answers.

ResultShow.startAnimation(AnimationUtils.loadAnimation(getApplicationContext(),
android.R.anim.fade_in)); // showing with animation.
        break;
        case 6:
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // turn ON
plus/minus button ...
        Toast.makeText(getApplicationContext(), "Number One +/- Number
Two", Toast.LENGTH_SHORT).show(); // if we click + or - before number one.
        BuildingFormulaStr = "";
        mathing.zero(TwoNumInFormula, str); // set ZERO to the array that
save the numbers.
        c = ' '; // reset PLUS/MINUS
        Formula.setText(""); // reset
    }
}

```

```

        CharactercounterInFormula = IndexerForNumbers = key = 0; // reset
        timerFeatures.cancel(); // STOP ...
        if (md30.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 2)
            timerFeatures.start(); // start helper timer if we have the
second feature.
        // timerFeatures - if we have second feature so we run this TIMER
to check when timer is DONE which button we have to Flickering.
        break;
    case 0:
        SwitchClickOneNumber = SwitchClickTwoNumber = false; // turn ON
plus/minus button ...
        break;
    }
    score.setText("Score : " + scores); // score text
    if (mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c,
TwoNumInFormula) != 0 && mathing.Checking(CharactercounterInFormula,
IndexerForNumbers, c, TwoNumInFormula) != 2 &&
mathing.Checking(CharactercounterInFormula, IndexerForNumbers, c, TwoNumInFormula) != 5) {
        mathing.CreaterNumbers(FormulaNumArray, NumberRange);
        mathing.ShowTheDetails(res, num1, num2, num3, num4, num5, num6,
FormulaNumArray);
        BuildingFormulaStr = "";
        CharactercounterInFormula = IndexerForNumbers = 0;
        Formula.setText("");
        mathing.zero(TwoNumInFormula, str);
        c = ' ';
    }
    timerMath.cancel(); // OFF
    if (md30.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 2)
        timerFeatures.start(); // start helper timer if we have the second
feature.
    // timerFeatures - if we have second feature so we run this TIMER to check
when timer is DONE which button we have to Flickering.
    locks(true); // turn ON.
}
}
public class CounterCurrentAnswer extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
and {@link #onFinish()}
     *
     * @param countDownInterval   The interval along the way to receive
     *                            {@link #onTick(long)} callbacks.
     */
    public CounterCurrentAnswer(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {
        long millis = millisUntilFinished;
        String hms = String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis), TimeUnit.MILLISECONDS.toMinutes(millis) -
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
TimeUnit.MILLISECONDS.toSeconds(millis) -
TimeUnit.MINUTES.getSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        // how the timer look like ...
    }
    // this is the function which run when the time is done !
}

```

```

@Override
public void onFinish() {
    mathing.CreaterNumbers(FormulaNumArray, NumberRange); // AGAIN we create
new numbers randomly ...
    mathing.ShowTheDetails(res, num1, num2, num3, num4, num5, num6,
FormulaNumArray); // // put all the values on the buttons after CREATING NUMBERS.
    BuildingFormulaStr = ""; // reset
    CharactercounterInFormula = IndexerForNumbers = 0; // reset
    Formula.setText(""); // reset
    mathing.zero(TwoNumInFormula, str); // set ZERO to the array that save the
numbers.
    c = ' '; // reset PLUS/MINUS buttons ...
    ResultShow.setText(""); // reset result show text ...
    timerFeatures.cancel();
    my_db md40 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
CALLING DB ...
    if(md40.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
        timerFeatures.start(); // start helper timer if we have the second
feature.
        // timerFeatures - if we have second feature so we run this TIMER to check
when timer is DONE which button we have to Flickering.
        locks(true); // call locks function.
    }
}
// locks function - lock/unlock buttons.
public void locks(boolean bn)
{
    // bn variable can be OR true OR false ...
    num1.setEnabled(bn);
    num2.setEnabled(bn);
    num3.setEnabled(bn);
    num4.setEnabled(bn);
    num5.setEnabled(bn);
    num6.setEnabled(bn);
}
}

```

קובץ : MathClass.java

מחלקה עזר למשחק ה - MATH

```
package com.example.fares.taguapp;

import android.content.DialogInterface;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.lang.*;
import java.lang.Math;
import java.util.Random;

/**
 * Created by Jamil-N on 3/11/2016.
 */
public class MathClass {
    public int H;
    public void CreterNumbers(int [] num,int con) // CreterNumbers creates the
matrix number randomly
    {
        // MUST TO HAVE AT LEAST ONE CORRECT RESULT !!
        Random rand = new Random();
        do {
            num[0] = rand.nextInt(con) + 1;
            num[1] = rand.nextInt(con) + 1;
            num[2] = rand.nextInt(con) + 1;
            num[3] = rand.nextInt(con) + 1;
            num[4] = rand.nextInt(con) + 1;
        }
        while (num[0] + num[0] != num[4] && num[0] + num[1] != num[4] && num[0] +
num[2] != num[4] && num[0] + num[3] != num[4] &&
        num[1] + num[1] != num[4] && num[1] + num[2] != num[4] && num[1] +
num[3] != num[4] &&
        num[2] + num[2] != num[4] && num[2] + num[3] != num[4] && num[3] +
num[3] != num[4] &&
        num[0] - num[0] != num[4] && num[0] - num[1] != num[4] && num[0] -
num[2] != num[4] && num[0] - num[3] != num[4] &&
        num[1] - num[1] != num[4] && num[1] - num[2] != num[4] && num[1] -
num[3] != num[4] &&
        num[2] - num[2] != num[4] && num[2] - num[3] != num[4] && num[3] -
num[3] != num[4]);
        H=num[4]; // SAVE THE RESULT because we will use it as soon as.
    }
    public String ShowResults(int [] num) // showing all the results
    {
        String stringres=""; // converting numbers result to text ...
        int c1=0,c2=0;
        for (int y = 0; y < 4; y++) {
            for (int k = 0; k < 4; k++) {
                if (num[y] + num[k] == num[4] && c1 == 0)
                {
                    stringres = stringres + Integer.toString(num[y]) + "+" +
Integer.toString(num[k]) + "=" + Integer.toString(num[4]) + "\n";
                    c1++; // just once - do not repeat.
                }
            }
        }
    }
}
```

```

        if (num[y] - num[k] == num[4] && c2 == 0)
        {
            stringres = stringres + Integer.toString(num[y]) + "—" +
Integer.toString(num[k]) + "=" + Integer.toString(num[4]) + "\n";
            c2++; // just once - do not repeat.
        }
        if (num[k] - num[y] == num[4] && c2 == 0)
        {
            stringres = stringres + Integer.toString(num[k]) + "—" +
Integer.toString(num[y]) + "=" + Integer.toString(num[4]) + "\n";
            c2++; // just once - do not repeat.
        }
    }
    return stringres;
}
public void ShowTheDetails(TextView r, Button b1, Button b2, Button b3, Button
b4, Button b5, Button b6, int [] n) // show details
{
    // show what we have after running CreateNumbers function.
    int i;
    char a6='+',a7='-' ; // just plus and minus.
    r.setText("The Result :" + Integer.toString(n[4]));
    b1.setText(Integer.toString(n[0]));
    b2.setText(Integer.toString(n[1]));
    b3.setText(Integer.toString(n[2]));
    b4.setText(Integer.toString(n[3]));
    b5.setText(String.valueOf(a6));
    b6.setText(String.valueOf(a7));
}
public int Checking(int count,int index,char charr,int [] n) // checking our
situation in the game.
{
    /* 1. in PLUS situation can return 1 or 2 -> 1 if we have the correct answer
OR 2 if we have the WRONG answer.
    2. in MINUS situation can return 3 or 4 or 5 :
    if 3 -> if we have correct answer and number one > number two
    if 4 -> if we have correct answer and number two > number one
    if 5 -> if we have the WRONG answer.
    3.we can also return 6 IF we did not finish builing the solution yet.
*/
    if (count == 3 && index == 2) {
        if (charr == '+') {
            if (n[0] + n[1] == H)
                return 1;
            else return 2;
        } else if (charr == '-') {
            if (n[0] > n[1] && n[0] - n[1] == H)
                return 3;
            else if (n[1] > n[0] && n[1] - n[0] == H)
                return 4;
            else return 5;
        }
        else return 6;
    }
    return 0;
}
public void zero(int[] n, String[] s) // ZERO function - To Reset the numbers and
strings array.
{
    for (int i=0;i<n.length;i++)
        n[i]=0;
    for (int i=0;i<s.length;i++)

```

```

        s[i]="";
    }
    // now we have all this functions - we create this functions for the features in
this GAME.
    public char ShowTheSecondCharacter(int [] num) // ShowTheSecondCharacter function
- to show the +/- CHARACTER when we use the flickering (HivHoov) Feature.
{
    int c1 = 0, c2 = 0;
    for (int y = 0; y < 4; y++) {
        for (int k = 0; k < 4; k++) { /* here we check which situation we have :
if PLUS so PLUS button is flickering
        if MINUS so MINUS button is flickering
        */
            if (num[y] + num[k] == num[4] && c1 == 0)
                return '+';
            if (num[y] - num[k] == num[4] && c2 == 0)
                return '-';
            if (num[k] - num[y] == num[4] && c2 == 0)
                return '-';
        }
    }
    return ' ';
}
public int ShowTheFirstCharacter(int [] num) // ShowTheFirstCharacter function -
to flickering the first CHARACTER (numberONE) when we use the flickeringFeature.
{
    int c1=0,c2=0;
    for (int y = 0; y < 4; y++) {
        for (int k = 0; k < 4; k++) { // checking if we are at PLUS situation or
MINIS situation,ACCORDING TO CREATENUMBER FUNCTION.
            if (num[y] + num[k] == num[4] && c1 == 0)
                return num[y];
            if (num[y] - num[k] == num[4] && c2 == 0)
                return num[y];
            if(num[k] - num[y] == num[4] && c2 == 0)
                return num[k];
        }
    }
    return -1; // for the code.
}
public int ShowTheThirdCharacter(int [] num) //ShowTheThirdCharacter function -
to flickering the Third CHARACTER (numberTWO) when we use the flickeringFeature.
{
    int c1=0,c2=0;
    for (int y = 0; y < 4; y++) {
        for (int k = 0; k < 4; k++) { // checking if we are at PLUS situation or
MINIS situation,ACCORDING TO CREATENUMBER FUNCTION.
            if (num[y] + num[k] == num[4] && c1 == 0)
                return num[k];
            if (num[y] - num[k] == num[4] && c2 == 0)
                return num[k];
            if(num[k] - num[y] == num[4] && c2 == 0)
                return num[y];
        }
    }
    return -1;
}
public void checkingthefirst(int s,Button num1,Button num2,Button num3,Button
num4) // who is the first in the flickering.
{
    // we get the four numbers in the matrix to check who the first/second number
to flickering..
    int u1=Integer.parseInt(num1.getText().toString());

```

```

        int u2=Integer.parseInt(num2.getText().toString());
        int u3=Integer.parseInt(num3.getText().toString());
        int u4=Integer.parseInt(num4.getText().toString());
        if(u1==s)
        {
            AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f);
            animation1.setDuration(800); // set duration...
            animation1.setStartOffset(1000); //setOffset ...
            num1.startAnimation(animation1); // runnning ...
            animation1.setRepeatCount(1); // just once ...
        }
        else if(u2==s)
        {
            AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f);
            animation1.setDuration(800); // set duration...
            animation1.setStartOffset(1000); //setOffset ...
            num2.startAnimation(animation1); // runnning ...
            animation1.setRepeatCount(1); // just once ...
        }
        else if(u3==s)
        {
            AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f);
            animation1.setDuration(800); // set duration...
            animation1.setStartOffset(1000); //setOffset ...
            num3.startAnimation(animation1); // runnning ...
            animation1.setRepeatCount(1); // just once ...
        }
        else if(u4==s)
        {
            AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f);
            animation1.setDuration(800); // set duration...
            animation1.setStartOffset(1000); //setOffset ...
            num4.startAnimation(animation1); // running ...
            animation1.setRepeatCount(1); // just once ...
        }
    }
    public void checkingthesecond(char s,Button num5,Button num6) // checkingthesecond
function - to run animation for the flickering
{
    if('+']==s)
    {
        AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f); // Alpha
OBJECT
        animation1.setDuration(800); // set duration
        animation1.setStartOffset(1000); // setting startOffset
        num5.startAnimation(animation1); // running
        animation1.setRepeatCount(1); // just once ...
    }
    else if('-']==s)
    {
        AlphaAnimation animation1 = new AlphaAnimation(0.0f, 1.0f); // Alpha
OBJECT
        animation1.setDuration(800); // set duration
        animation1.setStartOffset(1000); // setting startOffset
        num6.startAnimation(animation1); // running
        animation1.setRepeatCount(1); // just once ...
    }
}
}

```

קובץ typing.java

```
package com.example.fares.taguapp;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class typing extends AppCompatActivity {
    Button CurrentWordButton,AddWords; // buttons
    EditText EnterText; // editview
    TextView scoring; // textview
    List<String> WordsList = new ArrayList<String>(); // Words List
    int index,scoretype=0,AnimationDuration=7000,indexLoop;
    public int y=0;
    Random rand=new Random(); // rand VAR
    TranslateAnimation TranslateAnimating; // Animation VAR
    String
half="";ShershoorStringADDWORD="",NewOneString="",GetterWordsTyping="",CurrentWord,
HelperString;
    String[] WordsArray=new String[]{"shalom","friend","cow","ham","hanuka","hard
sauce","harvest","heart","hland","heaven","heritage","holding","holiday","holly","holy
","home","ward","hope","host","hostess",
" hugs","humility","hymns","magi","mail","majesty","manager","mantle","meat","and","spi
ces","memories","menorah","merry","messages","midnight",
"mincemeat","minister","miracle","miraculous","mistletoe","mittens","mosque","mrs","cl
aus","jessica","music"};;
    // ADDWORD function - To Add New Words to WordsList && we've the '*' between two
words in the list.
    public void ADDWORD()
    {
        final my_db md855 = new my_db(getApplicationContext(), "JAMIL", null, 1); // 
CALLING DB ...
        final AlertDialog.Builder C = new AlertDialog.Builder(typing.this); // Alert
Dialog
        C.setTitle("Please enter a word"); // enter a word ...
        final EditText in50 = new EditText(typing.this); // final to variables to make
its values not to be changed.
        C.setView(in50);
        C.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int whichButton) {
                WordsList.add(in50.getText().toString());
            }
        });
    }
}
```

```

ShershoorStringADDWORD=ShershoorStringADDWORD+in50.getText().toString()+"*"; // 
shershoor ...
        md855.updateWordsTyping(Signin.etName1.getText().toString(),
ShershoorStringADDWORD);
        WordsArray[WordsArray.length - 1] = in50.getText().toString(); // save
it at words array ...
        ADDAWORD(); // one more word ...
    }
});
C.setNegativeButton("CANCEL", null);
C.create().show();
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_typing);
    for (indexLoop = 0; indexLoop < WordsArray.length; indexLoop++)
        WordsList.add(WordsArray[indexLoop]); // adding to the list all the words
...
    final my_db md80 = new my_db(getApplicationContext(), "JAMIL", null, 1); //
calling DB ...
    if(md80.getWordsTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())!=null)
    {
        GetterWordsTyping =
md80.getWordsTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()); // get words
        for (indexLoop = 0; indexLoop < GetterWordsTyping.length(); indexLoop++)
{
            if (GetterWordsTyping.charAt(indexLoop) != '*')
                NewOneString += GetterWordsTyping.charAt(indexLoop);
            else
{
                WordsList.add(NewOneString); // Re-Build the NewWordString and
save it in the Words List ...
                NewOneString = ""; // for the next word ...
}
}
}
CurrentWordButton=(Button)findViewById(R.id.b); // button declarations ...
AddWords=(Button)findViewById(R.id.addwords); // button declarations ...
scoring=(TextView)findViewById(R.id.CUBESCORE); // textView declarations ...
scoretype=
md80.getScoreType(Signin.etName1.getText().toString(),Signin.etPass1.getText().toStrin
g()); // get type SCORE
    scoring.setText("Score : " + Integer.toString(scoretype));
EnterText = (EditText)findViewById(R.id.etType); // edittext declarations ...
index=rand.nextInt(WordsList.size()); // show it randomly ...
    CurrentWordButton.setText(WordsList.get(index)); // current word
button ...
    half = WordsList.get(index).substring(0, WordsList.get(index).length()/2); // 
HALF WORD we have to use it in specific feature ...
    final my_db md200 = new my_db(getApplicationContext(), "JAMIL", null, 1); //
calling DB...
    // To get focus and show the keyboard ...
    if(EnterText.requestFocus()) // FOCUS & SHOW the tablet keyboard ...
getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_VISIBLE);
    if(md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1
        {// if we've first feature - the application automatically write HALF WORD.

```

```

        EnterText.setText(half);
        EnterText.setSelection(WordList.get(index).length()/2); // set select
after the last character in HALF WORD.
    }
    else if(md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
    {   // if we've second feature - the application automatically write HALF WORD
& set new Animation Duration ...
        EnterText.setText(half);
        EnterText.setSelection(WordList.get(index).length()/2); // set select
after the last character in HALF WORD.
        AnimationDuration=16000;
    }
else
    EnterText.setText(""); // reset to next word ...
half="";
TranslateAnimating = new TranslateAnimation(0, 850, 0, 0); // set new
TranslateAnimation Object ...
TranslateAnimating.setDuration(AnimationDuration); // set duration
CurrentWordButton.startAnimation(TranslateAnimating); // start animation
TranslateAnimating.setAnimationListener(new Animation.AnimationListener())
{
    @Override
    public void onAnimationStart(Animation animation) {
    }
    // we animation ended ...
    @Override
    public void onAnimationEnd(Animation animation) {
        index = rand.nextInt(WordList.size()); // choose a word randomly
...
        CurrentWordButton.setText(WordList.get(index).toString()); // show the word on the button ...
        CurrentWordButton.startAnimation(TranslateAnimating); // starting animation ...
        EnterText.setText(""); // reset ..
        half=""; // reset for next word THEN ...
        half = WordList.get(index).substring(0, WordList.get(index).length()
/ 2); // HALF WORD for current word ...
        if (md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) >= 1)
        {
            // if we have - NOW - feature ...
            EnterText.setText(half); // put the HALF WORD
            EnterText.setSelection(WordList.get(index).length() / 2); // set select after the last character in HALF WORD.
        }
    }
    @Override
    public void onAnimationRepeat (Animation animation) {
}
});
EnterText.setOnLongClickListener(new View.OnLongClickListener()
{
    @Override
    public boolean onLongClick(View v) // Long Click
    {
        final String[] items = {"normal", "bold",
"italic", "bold & italic"}; // you can here change the FONT
        AlertDialog.Builder builder = new
AlertDialog.Builder(typing.this);
        builder.setTitle("Make your selection");
        builder.setItems(items, new

```

```

DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int item) {
        EnterText.setTypeface(EnterText.getTypeface(), Typeface.NORMAL); // normal
        if (items[0] == items[item]) {
            EnterText.setTypeface(EnterText.getTypeface(), Typeface.BOLD); // bold font
        } else if (items[1] == items[item]) {
            EnterText.setTypeface(EnterText.getTypeface(), Typeface.ITALIC); // italic font
        } else if (items[2] == items[item]) {
            EnterText.setTypeface(EnterText.getTypeface(), Typeface.BOLD_ITALIC); // bold_italic
            font
        }
    });
    EnterText.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int count, int after) {
        }
        @Override
        public void onTextChanged(CharSequence s, int start, int before, int count) {
            // here - we are checking if WHAT WE WROTE = TO
            // WHAT WE HAVE.
        }
        @Override
        public void afterTextChanged(Editable s) {
            HelperString = s.toString(); // helper
            CurrentWord =
            CurrentWordButton.getText().toString(); // get current word ...
            if (HelperString.contains(CurrentWord)) {
                my_db md79 = new
                my_db(getApplicationContext(), "JAMIL", null, 1); // calling DB ...
                scoretype++; // SCORE ++
                if (scoretype >= Store.updatetyping &&
                    md79.getStoreflagTyping(Signin.etName1.getText().toString(),
                    Signin.etPass1.getText().toString()) == 0)
                    Toast.makeText(getApplicationContext(), "you can go to the Store and buy the Half Word
                    Feature!", Toast.LENGTH_SHORT).show();
                else if (scoretype >= Store.updatetyping
                    && md79.getStoreflagTyping(Signin.etName1.getText().toString(),
                    Signin.etPass1.getText().toString()) == 1)
                    Toast.makeText(getApplicationContext(), "you can go to the Store and buy the Slow
                    Button Feature!", Toast.LENGTH_SHORT).show();
                md79.updaterTypeScore(Signin.etName1.getText().toString(), scoretype);
                scoring.setText("Score : " +
            }
        }
    });
}

```


קובץ letters.java

```
package com.example.fares.taguapp;

import android.os.Handler;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Locale;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;

public class letters extends AppCompatActivity {
    Button changeletter,again,TAGUWife; // buttons ...
    TextView catcher; // textviews
    TextToSpeech TextToSpeech; // text to speech
    int PhotoIndexer=0,randletter,please=0,ForTextToSpeech,delay = 120,period = 1500;
    // delay for 1 sec & repeat every 4 sec.
    Random rand=new Random(); // rand object
    String [] letterZ = new
    String[]{"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S",
    "T","U","V","W","X","Y","Z"};
    int[] PhotosOfLetterArray=new
    int[]{R.drawable.aa,R.drawable.bb,R.drawable.cc,R.drawable.dd,R.drawable.ee,R.drawable.
    ff,R.drawable.gg,R.drawable.hh,R.drawable.ii
    ,R.drawable.jj,R.drawable.kk,R.drawable.ll,R.drawable.mm,R.drawable.nn,R.drawable.oo,R
    .drawable.rr
    ,R.drawable.ss,R.drawable.tt,R.drawable.uu,R.drawable.vv,R.drawable.ww,R.drawable.xx,R
    .drawable.yy,R.drawable.zz};
    ImageView ScreenShow; // image view
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_letters);
        TAGUWife=(Button)findViewById(R.id.wife); // button declaration ...
        again=(Button)findViewById(R.id.btnHomePage); // button declaration ...
        changeletter=(Button)findViewById(R.id.chngletter); // button declaration ...
        my_listener m1 = new my_listener();
        again.setOnClickListener(m1);
        TAGUWife.setOnClickListener(m1);
        ScreenShow=(ImageView)findViewById(R.id.ivletters); // ImageView declaration
    ...
        catcher=(TextView)findViewById(R.id.txtcatch); // TextView declaration ...
        my_db md78 = new my_db(getApplicationContext(),"JAMIL", null, 1); // CALLING
        DB ...
        randletter=rand.nextInt(26); // get random letter index ...
        please = md78.getScoreLetter(Signin.etName1.getText().toString(),
        Signin.etPass1.getText().toString()); // get last score updated ...
        catcher.setText("Click the letter : " +letterZ[randletter] + "\n"+ "Score : " +
        please); // print last score updated ...
    }
}
```

```

        changeletter.setOnClickListener(new View.OnClickListener() {
            // if the user wants to change the letter -> he can click the
            "changeletter" button ...
            @Override
            public void onClick(View v) {
                randletter=rand.nextInt(26); // get random letter index ...
                catcher.setText("Click the letter : " +letterZ[randletter] +
"\n"+Score : " + please); // print last score updated ...
            }
        });
        ScreenShow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {   // Comparing two drawables ...
                // when we click - we check if what we have on screen show clicking
                EQUALS to the random letter index in PhotosOfLettersArray ...
                if
                (ScreenShow.getDrawable().getConstantState().equals(getResources().getDrawable(PhotosO
fLetterArray[randletter]).getConstantState()))
                    my_db md77 = new my_db(getApplicationContext(), "JAMIL", null, 1);
                // CALLING DB ...
                    please++; // score ++
                    md77.updateLetterScore(Signin.etName1.getText().toString(),
                please); // updating the new score in DB ...
                    Toast.makeText(getApplicationContext(), "GOOD ONE!",
                Toast.LENGTH_SHORT).show(); // greatfull message ...
                    randletter = rand.nextInt(26); // again -> get random letter index
                ...
            }
            catcher.setText("Click the letter : " + letterZ[randletter] + "\n" +
"Score : " + please); // print last score updated ...
        });
        // Handler - allows you to send and process Message and Runnable objects
        associated with a thread's MessageQueue.
        final Handler mHandler = new Handler();
        // Create runnable for posting.
        final Runnable mUpdateResults = new Runnable() {
            public void run()
            AnimateandSlideShow(); // AnimateandSlideShow function - set letters
            show in a specific range ...
        };
        Timer timer = new Timer(); // Timer - a single background thread that is used
        to execute all of the timer's tasks, sequentially.
        timer.scheduleAtFixedRate(new TimerTask()
        {
            public void run()
            mHandler.post(mUpdateResults); // mUpdateResults Runnable ->
            associated with a thread's MessageQueue.
        },
        delay, period); // Schedules the specified task for repeated fixed-rate
        execution, beginning after the specified delay.
    }
    class my_listener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case (R.id.btnAddPage):
                    SpeakerFunction(letterZ[PhotoIndexer]); // it tells us what is the
                    name of this letter on the screen show ...
                    break;
                case (R.id.wife):

```

```

        SpeakerFunction(letterZ[randletter]); // it tells us what is the
name of the letter that should click it ...
        break;
    }
}
public void SpeakerFunction(final String str) // SpeakerFunction function - to
take care of TextToSpeech variable ...
{
    TextToSpeech = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if (status == TextToSpeech.SUCCESS) // if we can use TextToSpeech
Object or not ...
    {
        ForTextToSpeech = TextToSpeech.setLanguage(Locale.US); // set US
Accent ...
        // if not supported ..
        if (ForTextToSpeech == TextToSpeech.LANG_NOT_SUPPORTED ||
ForTextToSpeech == TextToSpeech.LANG_MISSING_DATA)
            Toast.makeText(getApplicationContext(), "Feature not supported
in your device !", Toast.LENGTH_SHORT).show();
        else
            TextToSpeech.speak(str, TextToSpeech.QUEUE_ADD, null); // speaking ...
    }
    else // if not supported
        Toast.makeText(getApplicationContext(), "Feature not supported in
your device !", Toast.LENGTH_SHORT).show();
    }
});
private void AnimateandSlideShow() // AnimateandSlideShow function - set letters
show in a specific range ...
{
    // we - here - put a specific range according to the random letter index ...
    if(randletter>=23) // if the random letter index is at the end of the array
then we take the last 4 letters in the array.
    PhotoIndexer = rand.nextInt((randletter+1)-(randletter-4))+(randletter-4);
    else if(randletter<2) // if the random letter index is at the first/second
place of the array then we take the first 4 letters in the array.
    PhotoIndexer = rand.nextInt((randletter+4)-(randletter))+randletter;
    else
        PhotoIndexer = rand.nextInt((randletter+3)-(randletter-2))+(randletter-2);
    // if its in the middle we take 3 places before and after the random letter index ...
    ScreenShow.setImageResource(PhotosOfLetterArray[PhotoIndexer]); // show the
letter ...
    Animation rotateimage = AnimationUtils.loadAnimation(getApplicationContext(),
android.R.anim.slide_in_left); // set kind animation ...
    ScreenShow.startAnimation(rotateimage); // staring animation ...
}
}

```

קובץ focus.java

```
package com.example.fares.taguapp;

import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import java.util.Random;
import java.util.concurrent.TimeUnit;

public class focus extends AppCompatActivity {
    Button br1, br2, br3, br4, br5, br6, br7, br8, br9, br10, br11; // buttons
    boolean[] flag = new boolean[10];
    int[] PhotosArray = new int[]{R.drawable.spong, R.drawable.mikiii,
    R.drawable.mini, R.drawable.clienticon, R.drawable.apple};
    int i, RandomPlace, RandomPhoto ,RandomPlaceHelper, SCORE = 0, SecondTime=1000;
    Random rand = new Random(); // rand variable.
    final CounterClass CheckPressingTimer = new CounterClass(SecondTime, 1000); // setting Timer ...
    @Override
    protected void onStop() {
        super.onStop();
        CheckPressingTimer.cancel(); // dispose the timer ...
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_focus);
        br1 = (Button) findViewById(R.id.rr1); // Buttons Declaration ...
        br2 = (Button) findViewById(R.id.rr2); // Buttons Declaration ...
        br3 = (Button) findViewById(R.id.rr3); // Buttons Declaration ...
        br4 = (Button) findViewById(R.id.rr4); // Buttons Declaration ...
        br5 = (Button) findViewById(R.id.rr5); // Buttons Declaration ...
        br6 = (Button) findViewById(R.id.rr6); // Buttons Declaration ...
        br7 = (Button) findViewById(R.id.rr7); // Buttons Declaration ...
        br8 = (Button) findViewById(R.id.rr8); // Buttons Declaration ...
        br9 = (Button) findViewById(R.id.rr9); // Buttons Declaration ...
        br10 = (Button) findViewById(R.id.rr10); // Buttons Declaration ...
        br11 = (Button) findViewById(R.id.button4); // Buttons Declaration ...
        my_listener m2 = new my_listener(); //
        br1.setOnClickListener(m2); // connect button to the class
        br2.setOnClickListener(m2); // connect button to the class
        br3.setOnClickListener(m2); // connect button to the class
        br4.setOnClickListener(m2); // connect button to the class
        br5.setOnClickListener(m2); // connect button to the class
        br6.setOnClickListener(m2); // connect button to the class
        br7.setOnClickListener(m2); // connect button to the class
        br8.setOnClickListener(m2); // connect button to the class
        br9.setOnClickListener(m2); // connect button to the class
        br10.setOnClickListener(m2); // connect button to the class
        my_db md85 = new my_db(getApplicationContext(), "JAMIL", null, 1); // calling db ...
        SCORE =
        md85.getScoreFocus(Signin.etName1.getText().toString(),Signin.etPass1.getText().toString()); // get last score update ...
        br11.setText(SCORE + " "); // score text ...
        onCreate2(); // calling onCreate2 function ...
    }
}
```

```

public void onCreate2() {
    if(SCORE>100)
        SecondTime=800; // if score > 100 -> secTime -> 800
    if(SCORE<200)
    {
        final CounterClass CheckPressingTimer = new CounterClass(SecondTime,
1000); // set new SecondTime ...
        RandomPlace = rand.nextInt(10); // put random place ...
        RandomPhoto = rand.nextInt(4); // show random photo ...
        for (i = 0; i < 10; i++) // we set where is the right place and where is
not ...
    {
        if (i != RandomPlace)
            flag[i] = false; // if its not the right place ...
        else
            flag[i] = true; // if it is.
    }
    if (flag[0] == true) // show photo if its the right place ...
        br1.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br1.setBackgroundColor(PhotosArray[4]);
    if (flag[1] == true) // show photo if its the right place ...
        br2.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br2.setBackgroundColor(PhotosArray[4]);
    if (flag[2] == true) // show photo if its the right place ...
        br3.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br3.setBackgroundColor(PhotosArray[4]);
    if (flag[3] == true) // show photo if its the right place ...
        br4.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br4.setBackgroundColor(PhotosArray[4]);
    if (flag[4] == true) // show photo if its the right place ...
        br5.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br5.setBackgroundColor(PhotosArray[4]);
    if (flag[5] == true) // show photo if its the right place ...
        br6.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br6.setBackgroundColor(PhotosArray[4]);
    if (flag[6] == true) // show photo if its the right place ...
        br7.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br7.setBackgroundColor(PhotosArray[4]);
    if (flag[7] == true) // show photo if its the right place ...
        br8.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br8.setBackgroundColor(PhotosArray[4]);
    if (flag[8] == true) // show photo if its the right place ...
        br9.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br9.setBackgroundColor(PhotosArray[4]);
    if (flag[9] == true) // show photo if its the right place ...
        br10.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br10.setBackgroundColor(PhotosArray[4]);
    CheckPressingTimer.start(); // timer ON ...
}
else // if SCORE >= 200
{
    SecondTime=1000; // set 1 SECOND ...
    final CounterClass CheckPressingTimer = new CounterClass(SecondTime,

```

```

1000); // update timer
    RandomPlaceHelper= rand.nextInt(10); // second place
    do {
        RandomPlace = rand.nextInt(10);
    }while (RandomPlace==RandomPlaceHelper); // two photos in the same place -
impossible !!
    RandomPhoto = rand.nextInt(4); // random photo ...
    for (i = 0; i < 10; i++) {
        if (i == RandomPlace || i==RandomPlaceHelper) // the right place ..
or the first or the second
            flag[i] = true;
        else
            flag[i] = false; // wrong place ...
    }
    if (flag[0] == true) // right place ?
        br1.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br1.setBackgroundColor(PhotosArray[4]);
    if (flag[1] == true) // right place ?
        br2.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br2.setBackgroundColor(PhotosArray[4]);
    if (flag[2] == true) // right place ?
        br3.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br3.setBackgroundColor(PhotosArray[4]);
    if (flag[3] == true) // right place ?
        br4.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br4.setBackgroundColor(PhotosArray[4]);
    if (flag[4] == true) // right place ?
        br5.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br5.setBackgroundColor(PhotosArray[4]);
    if (flag[5] == true) // right place ?
        br6.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br6.setBackgroundColor(PhotosArray[4]);
    if (flag[6] == true) // right place ?
        br7.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br7.setBackgroundColor(PhotosArray[4]);
    if (flag[7] == true) // right place ?
        br8.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br8.setBackgroundColor(PhotosArray[4]);
    if (flag[8] == true) // right place ?
        br9.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br9.setBackgroundColor(PhotosArray[4]);
    if (flag[9] == true) // right place ?
        br10.setBackgroundResource(PhotosArray[RandomPhoto]);
    else
        br10.setBackgroundColor(PhotosArray[4]);
    CheckPressingTimer.start(); // timer ON ...
}
}

public class CounterClass extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
     *                            is called.
     */
    and {@link #onFinish()}
}

```

```

        * @param countDownInterval The interval along the way to receive
        *                      {@link #onTick(long)} callbacks.
        */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    @Override
    public void onTick(long millisUntilFinished) {
        long millis = millisUntilFinished;
        String hms = String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis), TimeUnit.MILLISECONDS.toMinutes(millis) -
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
TimeUnit.MILLISECONDS.toSeconds(millis) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        // how the timer look like ...
    }
    // this is the function which run when the time is done !
    @Override
    public void onFinish() {
        CheckPressingTimer.cancel(); // OFF
        onCreate2(); // call the onCreate2 function ...
    }
}
class my_listener implements View.OnClickListener
{
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.rr1:
                check(1); // checking if this is the right place
                break;
            case R.id.rr2:
                check(2); // checking if this is the right place
                break;
            case R.id.rr3:
                check(3); // checking if this is the right place
                break;
            case R.id.rr4:
                check(4); // checking if this is the right place
                break;
            case R.id.rr5:
                check(5); // checking if this is the right place
                break;
            case R.id.rr6:
                check(6); // checking if this is the right place
                break;
            case R.id.rr7:
                check(7); // checking if this is the right place
                break;
            case R.id.rr8:
                check(8); // checking if this is the right place
                break;
            case R.id.rr9:
                check(9); // checking if this is the right place
                break;
            case R.id.rr10:
                check(10); // checking if this is the right place
                break;
        }
    }
}
public void check(int x) // check(x) function - checking if this is the right
place or not.

```

```
{  
    if (flag[x - 1] == true) // checking if this place is the right place  
    {  
        if(SCORE>=200) // this is the right place , now we are checking here if  
the score > 200 or score < 200  
        SCORE += 5; // +5  
    else  
        SCORE += 10; // +10  
    my_db md86 = new my_db(getApplicationContext(), "JAMIL", null, 1); //  
calling DB ...  
    md86.updateScoreFocus(Signin.etName1.getText().toString(), SCORE); //  
update new score ...  
    br11.setText(SCORE + " "); // score text ...  
    }  
}  
}
```

קובץ photoss.java

```
package com.example.fares.taguapp;

import android.app.Activity;
import android.content.Intent;
import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.util.Random;
import java.util.concurrent.TimeUnit;

public class photoss extends AppCompatActivity {
    Button chance1, chance2, chance3; // buttons
    TextView SCORE, DownTimer; // textview
    ImageView ScreenShow; // imageview
    int Correctindex, RandomSort, timing, h4, k4, place, Indexer1Chance2, Indexer2Chance2;;
    static int score = 0; // player score.
    Random rindex = new Random(); // random variable
    CounterClass LevelTimer; // timer ...
    // A Photos Array for the pictures.
    int PhotosArray[] = new int[]{R.drawable.algeria, R.drawable.apple,
        R.drawable.argentina, R.drawable.baby, R.drawable.balloons, R.drawable.banana,
        R.drawable.basketball, R.drawable.bird, R.drawable.cat, R.drawable.books,
        R.drawable.coca_cola, R.drawable.dog, R.drawable.donkey, R.drawable.eagle,
        R.drawable.elephant, R.drawable.finland,
        R.drawable.france, R.drawable.germany, R.drawable.girl,
        R.drawable.gorillas,
        R.drawable.greece, R.drawable.headphones, R.drawable.israel,
        R.drawable.italy, R.drawable.ivorycoast,
        R.drawable.nature, R.drawable.monkey, R.drawable.one, R.drawable.orange,
        R.drawable.painter, R.drawable.bunny, R.drawable.fish, R.drawable.cow,
        R.drawable.kinder, R.drawable.snake, R.drawable.strawberry,
        R.drawable.eggplant, R.drawable.egg, R.drawable.chicken, R.drawable.nose};
    // photos description array for the photos
    String[] PhotosDescription = new String[]{"algeria", "apple", "argentina", "baby",
        "balloons", "banana", "basketball", "bird", "cat", "books", "coca_cola",
        "dog", "donkey", "eagle", "elephant", "finland", "france", "germany",
        "girl", "gorillas", "greece", "headphones",
        "israel", "italy", "ivorycoast", "nature", "monkey", "one", "orange",
        "painter", "bunny", "fish", "cow", "kinder", "snake", "strawberry",
        "eggplant", "egg", "chicken", "nose"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_photoss);
        chance1 = (Button) findViewById(R.id.chance1); // chances buttons and score
        text and the timer...
        chance2 = (Button) findViewById(R.id.chance2);
        chance3 = (Button) findViewById(R.id.chance3);
```

```

    SCORE = (TextView) findViewById(R.id.score5); // score text
    DownTimer = (TextView) findViewById(R.id.timerphoto); // timer text
    ScreenShow = (ImageView) findViewById(R.id.imageView2); // screen show view
    ...
    DownTimer.setText("00:00:04"); // timer down ...
    my_db md87 = new my_db(getApplicationContext(), "JAMIL", null, 1); // CALLING
DB ...
    score = md87.getScorePhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()); // get the last update score
    SCORE.setText("Score : " + score); // showing the score..
    // HERE -
    // we are checking some situations about the features in this GAME...
    if (md87.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 1)
    {
        // if we've the first feature - put 30 photos and 8 seconds.
        place = 30;
        timing = 8000;
    }
    else if (md87.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 2)
    {
        // if we've the second feature - put 40 photos and 8 seconds.
        timing = 8000;
        place = 40;
    }
    else // if we do not have any feature yet - put 30 photos and 4 seconds.
    {
        place = 30;
        timing = 4000;
    }
    Correctindex = rindex.nextInt(place); // we create randomly and save the
correct answer index.
    ScreenShow.setImageResource(PhotosArray[Correctindex]); // show photos of the
correct answer index.
    // we - here - have TWO Do-WHILE that creating randomly also TWO numbers for
TWO index's which not equal to the correct answer index.
    do {
        h4 = rindex.nextInt(place);
    } while (h4 == Correctindex); // if equal - create again.
    do {
        k4 = rindex.nextInt(place);
    } while (k4 == Correctindex || k4 == h4); // if equal - create again.
    // we sort the words on the button randomly .. its not static thing - we
change the words place.
    RandomSort = rindex.nextInt(3);
    switch (RandomSort) {
        case 0:
            // change places randomly..
            chance1.setText(PhotosDescription[Correctindex]);
            chance2.setText(PhotosDescription[h4]);
            chance3.setText(PhotosDescription[k4]);
            break;
        case 1:
            // change places randomly..
            chance3.setText(PhotosDescription[Correctindex]);
            chance2.setText(PhotosDescription[k4]);
            chance1.setText(PhotosDescription[h4]);
            break;
        case 2:
            // change places randomly..
            chance2.setText(PhotosDescription[Correctindex]);
            chance3.setText(PhotosDescription[k4]);
    }
}

```

```

        chance1.setText(PhotosDescription[h4]);
        break;
    }
    LevelTimer=new CounterClass(timing,1000); // we put time and run the timer...
    LevelTimer.start(); // if timer is done - it means we do not choose any
answer.
}
@Override
protected void onStop() {
    super.onStop();
    LevelTimer.cancel(); // when we exit the game - we dispose every thing.
}
public class CounterClass extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
     *                            and {@link #onFinish()} is called.
     * @param countDownInterval   The interval along the way to receive
     *                            {@link #onTick(long)} callbacks.
     */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {
        long millis = millisUntilFinished;
        String hms = String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis), TimeUnit.MILLISECONDS.toMinutes(millis) -
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
TimeUnit.MILLISECONDS.toSeconds(millis) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        DownTimer.setText(hms); // hms variable - how the timer look like when we
show it.
    }
    // this is the function which run when the time is done !
    @Override
    public void onFinish() {
        if (score != 0) // if score = 0 do nothing - ELSE : ->
        {
            my_db md150 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
CALLING DB ...
            score--; // lose one score
            Toast.makeText(getApplicationContext(), "you lose one score !",
Toast.LENGTH_SHORT).show();
            if (score >= Store.updatephotos &&
md150.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 0)
                Toast.makeText(getApplicationContext(), "you can go to the Store
and buy the +5 Seconds Feature!", Toast.LENGTH_SHORT).show();
            else if (score >= Store.updatephotos &&
md150.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 1)
                Toast.makeText(getApplicationContext(), "you can go to the Store
and buy the More Pictures Feature!", Toast.LENGTH_SHORT).show();
        }
        my_db md99 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
calling DATABASE ...
        md99.updaterScorePhotos(Signin.etName1.getText().toString(), score); // //
updateScorePhotos function - to update photos game score.
        SCORE.setText("Score : " + score); // our score.
        Correctindex = rindex.nextInt(place); // we create randomly and save the
    }
}

```

```

correct answer index.
    // we - here - have TWO Do-WHILE that creating randomly also TWO numbers
for TWO index's which not equal to the correct answer index.
    do {
        h4 = rindex.nextInt(place);
    } while (h4 == Correctindex); // if equal - create again ...
    do {
        k4 = rindex.nextInt(place);
    } while (k4 == Correctindex || k4 == h4); // if equal - create again ...
RandomSort = rindex.nextInt(3);
// we sort the words on the button randomly .. its not static thing - we
change the words place.
switch (RandomSort)
{
    case 0:
        // change places randomly..
        chance1.setText(PhotosDescription[Correctindex]);
        chance2.setText(PhotosDescription[h4]);
        chance3.setText(PhotosDescription[k4]);
        break;
    case 1:
        // change places randomly..
        chance3.setText(PhotosDescription[Correctindex]);
        chance2.setText(PhotosDescription[k4]);
        chance1.setText(PhotosDescription[h4]);
        break;
    case 2:
        // change places randomly..
        chance2.setText(PhotosDescription[Correctindex]);
        chance3.setText(PhotosDescription[k4]);
        chance1.setText(PhotosDescription[h4]);
        break;
}
ScreenShow.setImageResource(PhotosArray[Correctindex]); // show our new
correct answer index.
LevelTimer.cancel(); // OFF
LevelTimer.start(); // ON - if timer is done - it means we do not choose
any answer.
}
}
/* we have 3 choice buttons -
for every choice button we are checking if this button has the correct answer.
and every time we choose another words for the buttons randomly.
*/
public void b1(View v)
{
    if (v.getId() == R.id.chance1)
    {
        // if we press channce first button and its word is the same word for
correct answer words index then SCORE++ and update it.
        if
(chance1.getText().toString().contains(PhotosDescription[Correctindex].toString())) {
            score++; //Toast.makeText(getApplicationContext(),"GOOD
!", Toast.LENGTH_LONG).show();
            my_db md88 = new my_db(getApplicationContext(), "JAMIL", null, 1);
            md88.updateScorePhotos(Signin.etName1.getText().toString(),
score);
            if (score >= Store.updatephotos &&
md88.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 0)
                Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the +5 Seconds Feature!", Toast.LENGTH_SHORT).show();
            else if (score >= Store.updatephotos &&

```

```

md88.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 1)
    Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the More Pictures Feature!", Toast.LENGTH_SHORT).show();
    SCORE.setText("Score : " + score);
}
}
CreateAgain(); // CreateAgain() Function - create randomly two indexes for
wrong answer and put the words randomly on the buttons.
}
/* we have 3 choice buttons -
for every choice button we are checking if this button has the correct answer.
and every time we choose another words for the buttons randomly.
*/
public void b3(View v)
{
    if (v.getId() == R.id.chance3)
    {
        // if we press channce third button and its word is the same word for
correct answer words index then SCORE++ and update it.
        if
(chance3.getText().toString().contains(PhotosDescription[Correctindex].toString()))
{
            score++; //Toast.makeText(getApplicationContext(),"GOOD
!", Toast.LENGTH_LONG).show();
            my_db md89 = new my_db(getApplicationContext(), "JAMIL", null, 1);
            md89.updaterScorePhotos(Signin.etName1.getText().toString(),
score);
            if (score >= Store.updatephotos &&
md89.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 0)
                Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the +5 Seconds Feature!", Toast.LENGTH_SHORT).show();
            else if (score >= Store.updatephotos &&
md89.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() == 1)
                Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the More Pictures Feature!", Toast.LENGTH_SHORT).show();
            SCORE.setText("Score : " + score);
        }
    }
    CreateAgain(); // CreateAgain() Function - create randomly two indexes for
wrong answer and put the words randomly on the buttons.
}
/* we have 3 choice buttons -
for every choice button we are checking if this button has the correct answer.
and every time we choose another words for the buttons randomly.
*/
public void b2(View v)
{
    if (v.getId() == R.id.chance2)
    {
        // if we press channce second button and its word is the same word for
correct answer words index then SCORE++ and update it.
        if
(chance2.getText().toString().contains(PhotosDescription[Correctindex].toString()))
{
            score++; //Toast.makeText(getApplicationContext(),"GOOD
!", Toast.LENGTH_LONG).show();
            my_db md80 = new my_db(getApplicationContext(), "JAMIL", null, 1);
            md80.updaterScorePhotos(Signin.etName1.getText().toString(),
score);
            if (score >= Store.updatephotos &&
md80.getStoreflagPhotos(Signin.etName1.getText().toString(),

```

```

Signin.etPass1.getText().toString() == 0)
        Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the +5 Seconds Feature!", Toast.LENGTH_SHORT).show();
    else if (score >= Store.updatephotos &&
md80.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 1)
        Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the More Pictures Feature!", Toast.LENGTH_SHORT).show();
    SCORE.setText("Score : " + score);
}
}

CreateAgain(); // CreateAgain() Function - create randomly two indexes for
wrong answer and put the words randomly on the buttons.

}

public void CreateAgain() // CreateAgain() Function - create randomly two
indexes for wrong answer and put the words randomly on the buttons.
{
    Correctindex = rindex.nextInt(place); // we create randomly and save the
correct answer index.
    // we - here - have TWO Do-WHILE that creating randomly also TWO numbers for
TWO index's which not equal to the correct answer index.
    do {
        Indexer2Chance2 = rindex.nextInt(place);
    } while (Indexer2Chance2 == Correctindex);
    do {
        Indexer1Chance2 = rindex.nextInt(place);
    } while (Indexer1Chance2 == Correctindex || Indexer1Chance2 ==
Indexer2Chance2);
    RandomSort = rindex.nextInt(3);
    // we sort the words on the button randomly .. its not static thing - we
change the words place.
    switch (RandomSort)
    {
        case 0:
            // change places randomly..
            chance1.setText(PhotosDescription[Correctindex]);
            chance2.setText(PhotosDescription[Indexer2Chance2]);
            chance3.setText(PhotosDescription[Indexer1Chance2]);
            break;
        case 1:
            // change places randomly..
            chance3.setText(PhotosDescription[Correctindex]);
            chance2.setText(PhotosDescription[Indexer1Chance2]);
            chance1.setText(PhotosDescription[Indexer2Chance2]);
            break;
        case 2:
            // change places randomly..
            chance2.setText(PhotosDescription[Correctindex]);
            chance3.setText(PhotosDescription[Indexer1Chance2]);
            chance1.setText(PhotosDescription[Indexer2Chance2]);
            break;
    }
    ScreenShow.setImageResource(PhotosArray[Correctindex]); // show another photo
...
    LevelTimer.cancel(); // OFF
    LevelTimer.start(); // ON - if timer is done - it means we do not choose any
answer.
}
}

```

קובץ **cubelt.java**

LEVEL 1

```
package com.example.fares.taguapp;

import android.content.Intent;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Random;

public class cubeIt extends AppCompatActivity {
    Button movingtoabove, movingtowdown, switchOne; // buttons
    Button[] colors = new Button[4]; // buttons array
    TextView scoring; // textview
    Switch oner; // switcher
    int[] arr = {3, 2, 1, 0}; // every color has a specific number.
    int randomex,k=0;
    Random rand = new Random(); // random variable.
    boolean key=false;
    public void chancepositions() // chancepositions Fucntion - to change the
positions of buttons that depends on the movement CUBES.
    {
        /* two places - arr[0] And arr[2] - they are the places that we use them to
check the result when the animaiton ( collision ) is ended.
        arr[0] - for cubes which slide to above.
        arr[2] - for cubes which slide to down.
        every color has a specific number or 0 or 1 or 2 or 3
        0 red
        1 blue
        2 black
        3 green
        */
        int i,top0 = colors[0].getTop(),bot0 = colors[0].getBottom(),left0 =
colors[0].getLeft(),right0 = colors[0].getRight(),L = arr[0];
        // before the loop we saved the first item - we cant swap all the 4 items in
the loop because if we do it then we'll get a range error.
        for (i = 1; i <= 3; i++) // changing the position of buttons
        {
            colors[i - 1].setTop(colors[i].getTop()); // new top
            colors[i - 1].setBottom(colors[i].getBottom()); // new bottom
            colors[i - 1].setRight(colors[i].getRight()); // new right
            colors[i - 1].setLeft(colors[i].getLeft()); // new left
            arr[i - 1] = arr[i];
        }
        // the last = the first before the loop.
        colors[3].setTop(top0); // new top
```

```

        colors[3].setBottom(bot0); // new bottom
        colors[3].setRight(right0); // new right
        colors[3].setLeft(left0); // new left
        arr[i - 1] = L;
    }
    @Override
    protected void onStop()
    {
        super.onStop();
        finish(); // to end the Activity immediatly.
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cubedit);
        movingtowordown = (Button) findViewById(R.id.movetowordown); // setting findViewById
        movingtoabove = (Button) findViewById(R.id.movetoabove); // setting
        findViewById
        switchOne = (Button) findViewById(R.id.switcher); // setting findViewById
        colors[0] = (Button) findViewById(R.id.redmain); // setting findViewById
        colors[1] = (Button) findViewById(R.id.bluemain); // setting findViewById
        colors[2] = (Button) findViewById(R.id.blackmain); // setting findViewById
        colors[3] = (Button) findViewById(R.id.greenmain); // setting findViewById
        scoring = (TextView) findViewById(R.id.CUBESCORE); // setting findViewById
        oner=(Switch) findViewById(R.id.switcherone); // switcher from level to another
        level ...
        oner.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                key = true;
                Intent totwo = new Intent(getApplicationContext(), cubertwo.class);
                startActivity(totwo); // go to the Cube Game - Level 2 ...
            }
        });
        final my_db md81 = new my_db(getApplicationContext(), "JAMIL", null, 1); // CALLING DB ...
        k=md81.getScoreCube(Signin.etName1.getText().toString(),
        Signin.etPass1.getText().toString()); // get the last updated score.
        switchOne.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                chancepositions(); // chancepositions Fucntion - to change the
                positions of buttons that depends on the movement CUBES.
            }
        });
        RandomllyRunner(); // RandomllyRunner Function - randomly we choose which
        cubes play now .. with random colors every time
    }
    public void SlideToAbove()
    {
        Animation slide = null; // animation variable
        slide = new TranslateAnimation(Animation.RELATIVE_TO_SELF, 0.0f,
            Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
            0.0f, Animation.RELATIVE_TO_SELF, -8.2f); // settings for slide to
        ABOVE - setting the end point
        slide.setDuration(2500); // 2.5 sec
        movingtoabove.setVisibility(View.VISIBLE); // show cubes which moving to above
        movingtowordown.setVisibility(View.INVISIBLE);
        movingtoabove.startAnimation(slide); // animation is starting
        slide.setAnimationListener(new Animation.AnimationListener()
        {
            @Override
            public void onAnimationStart(Animation animation) {

```

```

        movingtobottom.setVisibility(View.INVISIBLE); // hide cubes which moving
to down
    }
    @Override
    public void onAnimationRepeat(Animation animation) {
    }
    @Override
    public void onAnimationEnd(Animation animation) {
        if (randomex == arr[0]) // if its the same color ( according to the
numbers values)
    {
        k++; // one more score
        final my_db md812 = new my_db(getApplicationContext(), "JAMIL",
null, 1); // CALLING DB ...
        md812.updateScoreCube(Signin.etName1.getText().toString(), k); // update the score
        Toast.makeText(getApplicationContext(), "Yes! " +
Integer.toString(k), Toast.LENGTH_SHORT).show(); // well done message.
    }
    RandomlyRunner(); // RandomlyRunner Function - randomly we choose
which cubes play now .. with random colors every time
}
});
public void SlideToDown()
{
    Animation slide = null; // animation variable
    slide = new TranslateAnimation(Animation.RELATIVE_TO_SELF, 0.0f,
        Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
        0.0f, Animation.RELATIVE_TO_SELF, 8.3f); // settings for slide to DOWN
- setting the end point
    slide.setDuration(2500); // 2.5 sec
    movingtobottom.setVisibility(View.INVISIBLE); // hide cubes which moving to
above
    movingtobottom.setVisibility(View.VISIBLE); // show cubes which moving to down
    movingtobottom.startAnimation(slide); // starting ...
    slide.setAnimationListener(new Animation.AnimationListener()
    {
        @Override
        public void onAnimationStart(Animation animation) {
            movingtobottom.setVisibility(View.INVISIBLE); // hide cubes which
moving to above
        }
        @Override
        public void onAnimationRepeat(Animation animation) {
        }
        @Override
        public void onAnimationEnd(Animation animation) {
            if (randomex == arr[2]) // if its the same color ( according to the
numbers values)
        {
            k++; // one more score
            final my_db md813 = new my_db(getApplicationContext(), "JAMIL",
null, 1); // CALLING DB ...
            md813.updateScoreCube(Signin.etName1.getText().toString(), k); // update the score
            Toast.makeText(getApplicationContext(), "Score : " +
Integer.toString(k), Toast.LENGTH_SHORT).show(); // well done message
        }
        RandomlyRunner(); // RandomlyRunner Function - randomly we choose
which cubes play now .. with random colors every time
    }
});
}

```

```

    }

    public void changerColorsButton(Button b) // changerColorsButton Function - to
choose the colors for the cubes
    {
        randomex=rand.nextInt(4); // random variable
        switch(randomex)
        {
            case 0:
                b.setBackground(Color.RED); // red
                break;
            case 1:
                b.setBackground(Color.BLUE); // blue
                break;
            case 2:
                b.setBackground(Color.BLACK); // black
                break;
            case 3:
                b.setBackground(Color.GREEN); // green
                break;
        }
    }

    public void RandomllyRunner() // RandomllyRunner Function - randomly we choose
which cubes play now .. with random colors every time
    {
        int ForRand = rand.nextInt(2); // random variable
        switch (ForRand)
        {
            case 0:
                changerColorsButton(movingtodown); // changerColorsButton Function -
to choose the colors for the cubes
                    SlideToDown(); // SlideToDown Function is running .. cubes slide to
down
                break;
            case 1:
                changerColorsButton(movingtoabove); // changerColorsButton Function -
to choose the colors for the cubes
                    SlideToAbove(); // SlideToAbove Function is running .. cubes slide to
above
                break;
        }
    }
}

```

קובץ **cubertwo.java**

LEVEL 2

```
package com.example.fares.taguapp;

import android.content.Intent;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.Switch;
import android.widget.Toast;

import java.util.Random;

public class cubertwo extends AppCompatActivity {
    Switch twor; // Switcher
    boolean
switcher=false, keyYellowOne=true, keyYellowTwo=true, keyWhiteOne=false, keyWhiteTwo=false
; // boolean
    int
randerOne, randerTwo, updaterSCORE=0, topone, botone, leftone, rightone, toptwo, bottwo, lefttw
o, righttwo, topone1, botone1, leftone1, rightone1, toptwo1, bottwo1, lefttwo1, righttwo1;
    Random rand = new Random(); // random variable
    Button
AboveBallerOne, AboveBallerTwo, DownBallerOne, DownBallerTwo, yellowerone, yellowertwo, whit
erone, whitertwo; // buttons
    @Override
    protected void onStop() {
        super.onStop();
        finish(); // to end the Activity immediatly.
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cubertwo);
        AboveBallerOne = (Button) findViewById(R.id.AboveBallOne); // findViewById
        AboveBallerTwo= (Button) findViewById(R.id.AboveBallTwo); // findViewById
        DownBallerOne = (Button) findViewById(R.id.DownBallOne); // findViewById
        DownBallerTwo =(Button) findViewById(R.id.DownBallTwo); // findViewById
        yellowerone= (Button) findViewById(R.id.yellowOne); // findViewById
        yellowertwo= (Button) findViewById(R.id.yellowTwo); // findViewById
        whiterone= (Button) findViewById(R.id.whiteOne); // findViewById
        whitertwo= (Button) findViewById(R.id.whiteTwo); // findViewById
        twor=(Switch) findViewById(R.id.switchertwo); // findViewById
        //
        yellowerone.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                changerpositionOne(); // for this button we run changerpositionOne()
function
                // two functions this buttons can use because we need to switch the
position of the button
            }
        });
        whiterone.setOnClickListener(new View.OnClickListener() {
```

```

    @Override
    public void onClick(View v) {
        changerpositionOne(); // for this button we run changerpositionOne()
function
        // two functions this buttons can use because we need to switch the
position of the button
    }
});
yellowertwo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        changerpositionTwo(); // for this button we run changerpositionTwo()
function too -
        // two functions this buttons can use because we need to switch the
position of the button
    }
});
whitetwo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        changerpositionTwo(); // for this button we run changerpositionTwo()
function too -
        // two functions this buttons can use because we need to switch the
position of the button
    }
});
final my_db md810 = new my_db(getApplicationContext(), "JAMIL", null, 1); // calling DB ...
updateSCORE=md810.getScoreCube(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()); // get last score update ...
twor.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switcher = true; // turn on.
        Intent toone = new Intent(getApplicationContext(), cubeIt.class); // setting new intent to transfer to another page.
        startActivity(toone); // go back to the level 1 ...
    }
});
RandomllyRunner(); // RandomllyRunner Function - randomly we choose which
cubes play now .. with random colors every time
}
public void SlideToAbove()
{
    Animation slide = null; // animation variable
    slide = new TranslateAnimation(Animation.RELATIVE_TO_SELF, 0.0f,
        Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
        0.0f, Animation.RELATIVE_TO_SELF, -9.5f); // settings for slide to
ABOVE - setting the end point
    slide.setDuration(2500); // 2.5 sec
    AboveBallerOne.setVisibility(View.INVISIBLE); // hide first cube which moving
to above
    AboveBallerTwo.setVisibility(View.INVISIBLE); // hide second cube which moving
to above
    DownBallerOne.setVisibility(View.VISIBLE); // show first cube which moving to
down
    DownBallerTwo.setVisibility(View.VISIBLE); // show second cube which moving to
down
    DownBallerOne.startAnimation(slide); // One Cube animation is running
    DownBallerTwo.startAnimation(slide); // Two Cube animation is running
    slide.setAnimationListener(new Animation.AnimationListener()
{
    @Override

```

```

        public void onAnimationStart(Animation animation) {
            AboveBallerOne.setVisibility(View.INVISIBLE); // hide first cube which
moving to above
            AboveBallerTwo.setVisibility(View.INVISIBLE); // hide second cube
which moving to above
        }
        @Override
        public void onAnimationRepeat(Animation animation) {
        }

        @Override
        public void onAnimationEnd(Animation animation)
        {
            // we put up yellow button and down white button as a default
            /* we put booleans keys varialbes that :
            1. yellow keys for two up buttons is FALSE as a default
            2. white keys for two up buttons is TRUE as a default
            3. randerOne and randerTwo variables can be or 0 or 1
            0 yellow
            1 white
            so,we are - here - checking every situation we may have- Relative to
the upside!!!!!
            checking for every side if the moving cube is yellow and the cubes
switcher is yellow too
            OR if the moving cube is white and the cubes switcher is white too
            */
            if(randerOne==0 && keyYellowOne==false || randerOne==1 &&
keyWhiteOne==false) // the first side if its the same colors
            {
                if (randerTwo == 0 && keyYellowTwo == false || randerTwo == 1 &&
keyWhiteTwo == false) // the second side if its the same colors
                {
                    updatersCORE++; // one more score
                    final my_db md812 = new my_db(getApplicationContext(),
"JAMIL", null, 1); // CALLING DB ...
                    md812.updateScoreCubetwo(Signin.etName1.getText().toString(),
updatersCORE); // update the score
                    Toast.makeText(getApplicationContext(), "Score : " +
Integer.toString(updatersCORE), Toast.LENGTH_SHORT).show(); // well done message
                }
            }
            RandomllyRunner(); // RandomllyRunner Function - randommally we choose
which cubes play now .. with random colors every time
        }
    });
}
public void SlideToDown()
{
    Animation slide = null; // animation variable
    slide = new TranslateAnimation(Animation.RELATIVE_TO_SELF, 0.0f,
        Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
        0.0f, Animation.RELATIVE_TO_SELF, 7.8f); // settings for slide to DOWN
- setting the end point
    slide.setDuration(2500); // 2.5 sec
    DownBallerOne.setVisibility(View.INVISIBLE); // hide first cube which moving to
down
    DownBallerTwo.setVisibility(View.INVISIBLE); // hide second cube which moving
to down
    AboveBallerOne.setVisibility(View.VISIBLE); // show first cube which moving to
above
    AboveBallerTwo.setVisibility(View.VISIBLE); // show second cube which moving to
above
    AboveBallerOne.startAnimation(slide); // One Cube animation is running
}

```

```

AboveBallerTwo.startAnimation(slide); // Two Cube animation is rinning
slide.setAnimationListener(new Animation.AnimationListener()
{
    @Override
    public void onAnimationStart(Animation animation) {
        DownBallerOne.setVisibility(View.INVISIBLE); // hide first cube which
moving to down
        DownBallerTwo.setVisibility(View.INVISIBLE); // hide second cube which
moving to down
    }
    @Override
    public void onAnimationRepeat(Animation animation) {
    }
    @Override
    public void onAnimationEnd(Animation animation)
    {
        // we put up yellow button and down white button as a default
        /* we put booleans keys varialbes that :
        1. yellow keys for two up buttons is FALSE as a default
        2. white keys for two up buttons is TRUE as a default
        3. randerOne and randerTwo variables can be or 0 or 1
        0 yellow
        1 white
        so,we are - here - checking every situation we may have- Relative to
the downside!!!!!
        checking for every side if the moving cube is yellow and the cubes
switcher is yellow too
        OR if the moving cube is white and the cubes switcher is white too
        */
        if(randerOne==0 && keyYellowOne==true || randerOne==1 &&
keyWhiteOne==true) // the first side if its the same colors
        {
            if (randerTwo == 0 && keyYellowTwo == true || randerTwo == 1 &&
keyWhiteTwo == true) // the second side if its the same colors
            {
                updaterSCORE++; // one more score
                final my_db md812 = new my_db(getApplicationContext(), "JAMIL",
null, 1); // CALLING DB ...
                md812.updateScoreCubetwo(Signin.etName1.getText().toString(),
updaterSCORE); // update the score
                Toast.makeText(getApplicationContext(), "Score : " +
Integer.toString(updaterSCORE), Toast.LENGTH_SHORT).show(); // well done message
            }
        }
        RandomllyRunner(); // RandomllyRunner Function - randomly we choose
which cubes play now .. with random colors every time
    }
});
}
public void changerColorsBallOne(Button b)
{
    randerOne=rand.nextInt(2); // rand variable
    switch(randerOne)
    {
        case 0:
            b.setBackgroundColor(Color.YELLOW); // yellow
            break;
        case 1:
            b.setBackgroundColor(Color.WHITE); // white
            break;
    }
}
public void changerColorsBallTwo(Button b)

```

```

{
    randerTwo = rand.nextInt(2); // rand variable
    switch (randerTwo) {
        case 0:
            b.setBackgroundColor(Color.YELLOW); // yellow
            break;
        case 1:
            b.setBackgroundColor(Color.WHITE); // white
            break;
    }
}

/* RightKeysSwitcher && LeftKeysSwitcher - we use it because we need to switch
the buttons and boolean keys together
because its important to the direction of the buttons that slide to down or
above.
*/
public void RightKeysSwitcher() // RightKeysSwitcher Function - we switch the
first ( right ) side buttons
{
    if(keyYellowOne==true) // if yellow button is down
        keyYellowOne=false; // yellow button is up
    else
        keyYellowOne=true; // yellow button is down
    if(keyWhiteOne==true) // if white button is down
        keyWhiteOne=false; // white button is up
    else
        keyWhiteOne=true; // white button is down
}

public void LeftKeysSwitcher() // LeftKeysSwitcher Function - we switch the second
(left ) side buttons
{
    if(keyYellowTwo==true)// if yellow button is down
        keyYellowTwo=false; // yellow button is up
    else
        keyYellowTwo=true; // yellow button is down
    if(keyWhiteTwo==true)// if white button is down
        keyWhiteTwo=false; // white button is up
    else
        keyWhiteTwo=true; // white button is down
}

public void changerpositionOne() // changerpositionOne Fucntion - for the first
side of buttons we swap their positions.
{
    topone=yellowerone.getTop(); // save the yellow one top
    botone=yellowerone.getBottom(); // save the yellow one bottom
    leftone=yellowerone.getLeft(); // save the yellow one left
    rightone=yellowerone.getRight(); // save the yellow one right
    //
    topone1=whiterone.getTop(); // save the white one top
    botone1=whiterone.getBottom(); // save the white one bottom
    leftone1 = whiterone.getLeft(); // save the white one left
    rightone1 = whiterone.getRight(); // save the white one right
    //
    whiterone.setTop(topone); // here we are swapping the top
    whiterone.setBottom(botone); // here we are swapping the bottom
    whiterone.setLeft(leftone); // here we are swapping the left
    whiterone.setRight(rightone); // here we are swapping the right
    //
    yellowerone.setTop(topone1); // here we are swapping the top
    yellowerone.setBottom(botone1); // here we are swapping the bottom
    yellowerone.setLeft(leftone1); // here we are swapping the left
}

```

```

yellowerone.setRight(rightone1); // here we are swapping the right
RightKeysSwitcher(); // RightKeysSwitcher Function - we switch the first (
right ) side buttons
}

public void changerpositionTwo() // changerpositionTwo Fucntion - for the second
side of buttons we swap their positions.
{
    toptwo=yellowertwo.getTop(); // save the yellow two top
    bottwo=yellowertwo.getBottom(); // save the yellow two bottom
    lefttwo=yellowertwo.getLeft(); // save the yellow two left
    righttwo=yellowertwo.getRight(); // save the yellow two right
    //
    toptwol=whitertwo.getTop(); // save the yellow two top
    bottwol=whitertwo.getBottom(); // save the yellow two bottom
    lefttwol=whitertwo.getLeft(); // save the yellow two left
    righttwol=whitertwo.getRight(); // save the yellow two right
    //
    whitertwo.setTop(toptwo); // here we are swapping the top
    whitertwo.setBottom(bottwo); // here we are swapping the bottom
    whitertwo.setLeft(lefttwo); // here we are swapping the left
    whitertwo.setRight(righttwo); // here we are swapping the right
    //
    yellowertwo.setTop(toptwol); // here we are swapping the top
    yellowertwo.setBottom(bottwol); // here we are swapping the bottom
    yellowertwo.setLeft(lefttwol); // here we are swapping the left
    yellowertwo.setRight(righttwol); // here we are swapping the right
    LeftKeysSwitcher(); // LeftKeysSwitcher Function - we switch the second ( left
) side buttons
}

public void RandomllyRunner() // RandomllyRunner Function - randomly we choose
which cubes play now .. with random colors every time
{
    int ForRand = rand.nextInt(2); // rand variable
    switch (ForRand)
    {
        case 0:
            changerColorsBallOne(AboveBallerOne); // changerColorsBallOne Fucntion
            - to choose the colors for the First Abovecube
            changerColorsBallTwo(AboveBallerTwo); // changerColorsBallTwo Fucntion
            - to choose the colors for the Second Abovecube
            SlideToDown(); // SlideToDown Function is running .. cubes slide to
            down
            break;
        case 1:
            changerColorsBallOne(DownBallerOne); // changerColorsBallOne Fucntion
            - to choose the colors for the First Downcube
            changerColorsBallTwo(DownBallerTwo); // changerColorsBallTwo Fucntion
            - to choose the colors for the Second Downcube
            SlideToAbove(); // SlideToDown Function is running .. cubes slide to
            above
            break;
    }
}

```

קובץ colors.java

```
package com.example.fares.taguapp;

import android.media.MediaPlayer;
import android.os.CountDownTimer;
import android.speech.tts.TextToSpeech;
import android.speech.tts.Voice;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.Locale;
import java.util.Random;
import java.util.concurrent.TimeUnit;

public class colors extends AppCompatActivity {
    Button red, blue, green, again, yellow, black, white,skip; // buttons
    int[] names = new int[10];
    int[] randomnames = new int[10];
    String[] str = new String[10];
    int flag=0,counter = 0, j = 0, i = 0,forTextSpeech, score = 0, timing=2000,k=1,
    scoreHelper = 0;
    CounterClass timerToStart =new CounterClass(timing,1000); // timer to start for 1
second.
    CounterClass2 timerForWaitingYou =new CounterClass2(15000,1000); // timer for 15
seconds to ask if the user still there
    String StrForSpeech, StringForColorsString; // String
    TextView SCORETEXT, OURCOLOR; // textView
    TextToSpeech textToSpeech; // text to speech
    Toast toast; // toast message
    Random rand = new Random(); // rand variable
    my_listener myListener = new my_listener();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_colors);
        for (int i = 0; i < 10; i++)
        {
            names[i] = 7; // set all 7
            randomnames[i] = 7; // set all 7
        }
        toast=Toast.makeText(colors.this, "", Toast.LENGTH_SHORT); // set toast
    message
        red = (Button) findViewById(R.id.btnRED); // button declaration
        blue = (Button) findViewById(R.id.btnBLUE); // button declaration
        green = (Button) findViewById(R.id.btnGREEN); // button declaration
        again = (Button) findViewById(R.id.btnagaining); // button declaration
        yellow = (Button) findViewById(R.id.btnYELLOW); // button declaration
        white = (Button) findViewById(R.id.btnWHITE); // button declaration
        black = (Button) findViewById(R.id.btnBLACK); // button declaration
        skip = (Button) findViewById(R.id.skip); // button declaration
        SCORETEXT = (TextView) findViewById(R.id.t3); // textView declaration
        OURCOLOR = (TextView) findViewById(R.id.ourcolor); // textView declaration
        my_db md83 = new my_db(getApplicationContext(), "JAMIL", null, 1); // CALLING
DB ...
        score=md83.getScoreColors(Signin.etName1.getText().toString(),Signin.etPass1.getText())
    }
}
```

```

        .toString()); // get last update score ...
        if(md83.getStoreflagColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())>=1)
            skip.setEnabled(true); // turn ON if you have the first feature
        else
            skip.setEnabled(false); // turn OFF if you do not.
        SCORETEXT.setText("SCORE = " + score);
        myListener.creatingAndvoicing(); // creatingAndvoicing function - to create
randomly string to speak with specific number of colors.
        OURCOLOR.setText(StringForColorsString);
        black.setOnClickListener(myListener); // connect button the to class
        white.setOnClickListener(myListener); // connect button the to class
        yellow.setOnClickListener(myListener); // connect button the to class
        blue.setOnClickListener(myListener); // connect button the to class
        red.setOnClickListener(myListener); // connect button the to class
        green.setOnClickListener(myListener); // connect button the to class
        again.setOnClickListener(myListener); // connect button the to class
        skip.setOnClickListener(myListener); // connect button the to class
    }
    @Override
    protected void onStop ()
    {
        super.onStop();
        toast.cancel(); // dispose messages...
        timerForWaitingYou.cancel(); // dispose every thing
        timerToStart.cancel(); // dispose every thing
        textToSpeech.stop(); // dispose the speech
        finish(); //to end the Activity immediatly.
    }
    class my_listener implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case (R.id.btnagaining):
                    my_db md86 = new my_db(getApplicationContext(), "JAMIL", null,
1); // calling db ...

if(md86.getStoreflagColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
{
                red.setEnabled(false); // OFF
                black.setEnabled(false); // OFF
                white.setEnabled(false); // OFF
                yellow.setEnabled(false); // OFF
                blue.setEnabled(false); // OFF
                green.setEnabled(false); // OFF
                textToSpeech.speak(StrForSpeech, TextToSpeech.QUEUE_ADD,
null); // speaking ...
                timerToStart.start(); //
                timerForWaitingYou.cancel(); // OFF
                timerForWaitingYou.start(); // ON
}
else
{
                toast.setText("buy help from TAGU Store!"); // buy it from the
store
                toast.show(); // show it.
}
break;
case (R.id.skip): // pressing skip button
{
                my_db md833 = new my_db(getApplicationContext(), "JAMIL", null,
1); // calling DB ...
}
}
}

```

```

        if(md833.getStoreflagColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())>=1)
            creatingAndvoicing(); // if you have the first feature ..
        else
        {
            toast.setText("buy help from TAGU Store!");// buy it from the
store
            toast.show(); // show it.
        }
    }
break;
case (R.id.btnRED):
    if (i < k) { // while we are not at the end ...
        names[i] = 0;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
case (R.id.btnBLUE):
    if (i < k) { // while we are not at the end ...
        names[i] = 1;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
case (R.id.btnGREEN):
    if (i < k) { // while we are not at the end ...
        names[i] = 2;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
case (R.id.btnBLACK):
    if (i < k) { // while we are not at the end ...
        names[i] = 3;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
case (R.id.btnYELLOW):
    if (i < k) { // while we are not at the end ...
        names[i] = 4;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
case (R.id.btnWHITE):
    if (i < k) { // while we are not at the end ...
        names[i] = 5;
        flag=1; // if we click this button - we check it because the
app every 15 checks if we are still here or not ...
        check(i); // check it if correct/incorrect
        break;
    }
}
public void check(int j)
{

```

```

        if (!(names[j] == randomnames[j])) // if we click the wrong button ... not
according to the colors string ...
    {
        textToSpeech.shutdown(); // shut down the speech if we click wrong
button ( not according to the color string which we created )
        score-=10; // lose a score
        my_db md866 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
CALLING DB ...
        md866.updaterScoreColors(Signin.etName1.getText().toString(), score);
// update new score
        SCORETEXT.setText("SCORE" + " =" + " " + score); // score text
        toast.setText("You lose this round!"); // sorry message
        toast.show(); // show it ...
        i = counter = flag=0; // did not click any button yet ...
        OURCOLOR.setText("");
        creatingAndvoicing(); // creating randomly AGAIN new colors string
for next time ...
        OURCOLOR.setText(StringForColorsString);
    }
    else
    {
        counter++; // counter for the right colors orders in the string ... (
retseef mazbot )
        if (counter == k) // if we are at the end because K is the numbers of
colors ...
        {
            toast.setText("Good job !"); // correct answer
            toast.show(); // show message ...
            score += 10; // more score
            my_db md84 = new my_db(getApplicationContext(), "JAMIL", null, 1);
// CALLING DB ...

        md84.updaterScoreColors(Signin.etName1.getText().toString(), score); // update new
score
            if (score >= Store.updatecolors &&
        md84.getStoreflagColors(Signin.etName1.getText().toString(),
        Signin.etPass1.getText().toString()) == 0)
                Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the Skip Level Feature!", Toast.LENGTH_SHORT).show();
            else if (score >= Store.updatecolors &&
        md84.getStoreflagColors(Signin.etName1.getText().toString(),
        Signin.etPass1.getText().toString()) == 1)
                Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the Again Level Feature!", Toast.LENGTH_SHORT).show();
            i = counter = flag=0; //reset and set FLAG zero to check if we do
click any button in the next step ..
            scoreHelper += 10; // counting for 2 steps correct answers
            if (scoreHelper > 20 && k < 10) {
                scoreHelper = 0; // after two correct answers we add one more
color ...
                k++; // how many colors we have - current colors number ...
                toast.setText("One More Color"); // add one more color
                toast.show(); // show it ...
                if(score>50)
                    timing+=1000; // more 1 seconds if score > 50 ...
                timerToStart =new CounterClass(timing,400); // updating new
time for the timer ...
                SCORETEXT.setText("SCORE" + " =" + " " + score); // score text
...
            }
            SCORETEXT.setText("SCORE" + " =" + " " + score); // score text ...
            OURCOLOR.setText(""); // rest
            creatingAndvoicing(); // creating AGAIN new COLORS string
    }
}

```

```

randomally ...
    OURCOLOR.setText(StringForColorsString); // string colors text ...
}
else
    i++; // if we are not at the end of the string -> keep moving ...
}
flag=0; // set flag zero to check if we do click any button in the next
step ...
}
public void creatingAndvoicing() {
    StrForSpeech = " ";
    StringForColorsString = "";
    for (j = 0; j < k; j++) {
        int x = rand.nextInt(6);
        randomnames[j] = x; // randomly creating string - 6 colors ...
        switch (x)
        {
            case 0:
                str[j] = "Red"; // red
                break;
            case 1:
                str[j] = "blue"; // blue
                break;
            case 2:
                str[j] = "green"; // green
                break;
            case 3:
                str[j] = "BLACK"; // black
                break;
            case 4:
                str[j] = "YELLOW"; // yellow
                break;
            case 5:
                str[j] = "WHITE"; // white
                break;
        }
        StrForSpeech += "\"\" + str[j] + "\" \"";
    }
    StrForSpeech += "\"\" + str[j] + "\" \"";
    with '\' it makes the APP take more time to speak color after color ...
    StringForColorsString = StringForColorsString + str[j] + " ";
}
Shershoor ...
}
textToSpeech = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if (status == TextToSpeech.SUCCESS) {
            forTextSpeech = textToSpeech.setLanguage(Locale.US); // set US
ACCENT LANGUAGE ...
        if (forTextSpeech == TextToSpeech.LANG_NOT_SUPPORTED ||
forTextSpeech == TextToSpeech.LANG_MISSING_DATA) // if supported
            Toast.makeText(getApplicationContext(), "Feauter not
supported in your device !", Toast.LENGTH_SHORT).show();
        else {
            red.setEnabled(false); // OFF
            black.setEnabled(false); // OFF
            white.setEnabled(false); // OFF
            yellow.setEnabled(false); // OFF
            blue.setEnabled(false); // OFF
            green.setEnabled(false); // OFF
            textToSpeech.speak(StrForSpeech, TextToSpeech.QUEUE_ADD,
null); // speaking
            timerToStart.start(); // ON
            timerForWaitingYou.cancel(); // OFF
        }
    }
}

```

```

                timerForWaitingYou.start(); // ON
            }
        } else // not supported message ...
            Toast.makeText(getApplicationContext(), "Feature not supported
in your device !", Toast.LENGTH_SHORT).show();
        });
    }
}

public class CounterClass extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
     *                            and {@link #onFinish()} is called.
     * @param countDownInterval   The interval along the way to receive
     *                            {@link #onTick(long)} callbacks.
     */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {
        long millis =millisUntilFinished;
        String hms=String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis),TimeUnit.MILLISECONDS.toMinutes(millis)-
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
TimeUnit.MILLISECONDS.toSeconds(millis)-
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        // how timer look like ...
    }
    // this is the function which run when the time is done !
    @Override
    public void onFinish() {
        red.setEnabled(true); // ON
        black.setEnabled(true); // ON
        white.setEnabled(true); // ON
        yellow.setEnabled(true); // ON
        blue.setEnabled(true); // ON
        green.setEnabled(true); // ON
    }
}
//-----
// Are You Still There ??
public class CounterClass2 extends CountDownTimer {
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
     *                            and {@link #onFinish()} is called.
     * @param countDownInterval   The interval along the way to receive
     *                            {@link #onTick(long)} callbacks.
     */
    public CounterClass2(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {
        long millis =millisUntilFinished;
        String hms=String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis),TimeUnit.MILLISECONDS.toMinutes(millis)-

```

```

TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
    TimeUnit.MILLISECONDS.toSeconds(millis)-
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
        // how timer look like ...
    }
    // this is the function which run when the time is done !
    // here we check - for 15 seconds if we do not touch/click any thing on the
APP then the APP asks us if we are still here or not !
    // when we click - this timer is off and on agian ...
    @Override
    public void onFinish() {
        if(flag==0) // if we did not click any button for 15 seconds ...
            textToSpeech.speak("Are You Still There ??!!!",TextToSpeech.QUEUE_ADD,
null);
            timerForWaitingYou.start(); // ON
    }
}
}
}

```

קובץ clock.java

```
package com.example.fares.taguapp;

import android.content.Intent;
import android.media.MediaPlayer;
import android.os.CountDownTimer;
import android.speech.tts.TextToSpeech;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.RadioButton;
import android.widget.RelativeLayout;

import android.widget.TextView;
import android.widget.Toast;

import java.util.Locale;
import java.util.Random;
import java.util.concurrent.TimeUnit;
import java.util.logging.Handler;

public class clock extends AppCompatActivity {

    ImageButton
clock1,clock2,clock3,clock4,clock5,clock6,clock7,clock8,clock9,clock10,clock11,clock12
    ;
    RadioButton clockcenter;
    RelativeLayout r11;
    Button ResetAndTryAgain;
    int Linescounter=0,i,j,IndexClock,IndexMinute,SCORE=0;
    View k1,k2; // two view for building the CLOCK/MINUTE Line...
    final CounterClass Checktimer =new CounterClass(2000,1000);
    Random rand = new Random();
    TextView currentclock;
    TextToSpeech textToSpeech;
    MediaPlayer Mp;
    // An Oclock Array ...
    String[][] OclockArray={

"12:00","12:05","12:10","12:15","12:20","12:25","12:30","12:35","12:40","12:45","12:50
","12:55"},

{"01:00","01:05","01:10","01:15","01:20","01:25","01:30","01:35","01:40","01:45","01:5
0","01:55"},

{"02:00","02:05","02:10","02:15","02:20","02:25","02:30","02:35","02:40","02:45","02:5
0","02:55"},

{"03:00","03:05","03:10","03:15","03:20","03:25","03:30","03:35","03:40","03:45","03:5
0","03:55"},

{"04:00","04:05","04:10","04:15","04:20","04:25","04:30","04:35","04:40","04:45","04:5
0","04:55"},

{"05:00","05:05","05:10","05:15","05:20","05:25","05:30","05:35","05:40","05:45","05:5
0","05:55"},
```

```

        {"06:00", "06:05", "06:10", "06:15", "06:20", "06:25", "06:30", "06:35", "06:40", "06:45", "06:50", "06:55"},  

        {"07:00", "07:05", "07:10", "07:15", "07:20", "07:25", "07:30", "07:35", "07:40", "07:45", "07:50", "07:55"},  

        {"08:00", "08:05", "08:10", "08:15", "08:20", "08:25", "08:30", "08:35", "08:40", "08:45", "08:50", "08:55"},  

        {"09:00", "09:05", "09:10", "09:15", "09:20", "09:25", "09:30", "09:35", "09:40", "09:45", "09:50", "09:55"},  

        {"10:00", "10:05", "10:10", "10:15", "10:20", "10:25", "10:30", "10:35", "10:40", "10:45", "10:50", "10:55"},  

        {"11:00", "11:05", "11:10", "11:15", "11:20", "11:25", "11:30", "11:35", "11:40", "11:45", "11:50", "11:55"}  

    };  

    @Override  

    protected void onCreate(Bundle savedInstanceState) {  

        super.onCreate(savedInstanceState);  

        setContentView(R.layout.activity_clock);  

        ResetAndTryAgain = (Button) findViewById(R.id.rstryagain);  

        currentclock=(TextView)findViewById(R.id.txtClock);  

        clock1 = (ImageButton) findViewById(R.id.imageButton1); // Clocks Declaration  

        clock2 = (ImageButton) findViewById(R.id.imageButton2); // Clocks Declaration  

        clock3 = (ImageButton) findViewById(R.id.imageButton3); // Clocks Declaration  

        clock4 = (ImageButton) findViewById(R.id.imageButton4); // Clocks Declaration  

        clock5 = (ImageButton) findViewById(R.id.imageButton5); // Clocks Declaration  

        clock6 = (ImageButton) findViewById(R.id.imageButton6); // Clocks Declaration  

        clock7 = (ImageButton) findViewById(R.id.imageButton7); // Clocks Declaration  

        clock8 = (ImageButton) findViewById(R.id.imageButton8); // Clocks Declaration  

        clock9 = (ImageButton) findViewById(R.id.imageButton9); // Clocks Declaration  

        clock10 = (ImageButton) findViewById(R.id.imageButton10); // Clocks Declaration  

        Declaration  

        clock11 = (ImageButton) findViewById(R.id.imageButton11); // Clocks Declaration  

        Declaration  

        clock12 = (ImageButton) findViewById(R.id.imageButton12); // Clocks Declaration  

        Declaration  

        clockcenter = (RadioButton) findViewById(R.id.radioButton44);  

        final my_db md81 = new my_db(getApplicationContext(), "JAMIL", null, 1); //  

        CALLING DB...  

        SCORE=md81.getScoreClock(Signin.etName1.getText().toString(),Signin.etPass1.getText().  

        toString()); // get the last updated score  

        Mp=MediaPlayer.create(clock.this, R.raw.wrong); // we create the ringtone for  

        the wrong answer  

        textToSpeech = new TextToSpeech(getApplicationContext(), new  

        TextToSpeech.OnInitListener() {  

            @Override  

            public void onInit(int status) {  

                if (status == TextToSpeech.SUCCESS) {  

                    int res4 = textToSpeech.setLanguage(Locale.US); // set US accent -  

Lang.  

                    if (res4 == TextToSpeech.LANG_NOT_SUPPORTED || res4 ==  

TextToSpeech.LANG_MISSING_DATA) // if not support  

                        Toast.makeText(getApplicationContext(), "Feature not supported  

in your device !", Toast.LENGTH_SHORT).show();  

                } else  

                    Toast.makeText(getApplicationContext(), "Feature not supported in  

your device !", Toast.LENGTH_SHORT).show();  

            }  

        });
    }
}

```

```

r11 = (RelativeLayout) findViewById(R.id.r11);
i=rand.nextInt(12); // random line ( clock ) index
j=rand.nextInt(12); // random line ( column ) index
currentclock.setText("Write:" + OclockArray[i][j] + "\nScore : " + SCORE );
// o'clock and score
ResetAndTryAgain.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // reset all if we want to try again...
        Linescounter=0; // no lines.
        r11.removeView(k1); // remove clock line
        r11.removeView(k2); // remove any minute line
    }
});
clock1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2)
        {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock1); // line
for clock
                IndexClock=1; // clock on ONE
                if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
                {
                    if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1) {
                        textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                        r11.removeView(k1); // remove clock line
                        Linescounter=-1; // we remove the wrong line
                    }
                    else
                        r11.addView(k1); // add clock line
                }
                else
                    r11.addView(k1); // add clock line
            }
            else if(Linescounter==1) // if we have the second line
            {
                k2 = new MinuteLine(clock.this, clockcenter, clock1); // line
for minute
                IndexMinute=1; // mintue on ONE
                r11.addView(k2); // add minute line
                Checktimer.start(); // timer for checking the answer.
                locks(false); // cannot press any button while checking the
answer. // cannot press any button while checking the answer.
            }
            Linescounter++; // just 2 lines we need - no more and no less ...
// just 2 lines we need - no more and no less ...
        }
    }
});
clock2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2) {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock2); // line
for clock
                IndexClock=2; // clock on TWO
            }
        }
    }
});

```

```

        if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
    {
        if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
            textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
            r11.removeView(k1); // remove clock line
            Linescounter=-1; // we remove the wrong line
        }
        else
            r11.addView(k1); // add clock line
    }
    else
        r11.addView(k1); // add clock line
}
else if(Linescounter==1) // if we have the second line
{
    k2 = new MinuteLine(clock.this, clockcenter, clock2); // line
for minute
    IndexMinute=2; // mintue on TWO
    r11.addView(k2);
    Checktimer.start(); // timer for checking the answer.
    locks(false); // cannot press any button while checking the
answer.
}
Linescounter++; // just 2 lines we need - no more and no less ...
}
}
});
clock3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (Linescounter != 2) {
    if(Linescounter==0) // if we have the first line
    {
        k1 = new ClockLine(clock.this, clockcenter, clock3); // line
for clock
        IndexClock=3; // clock on THREE
        if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
        {
            if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
                textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                r11.removeView(k1); // remove clock line
                Linescounter=-1; // we remove the wrong line
            }
            else
                r11.addView(k1); // add clock line
        }
        else
            r11.addView(k1); // add clock line
    }
    else if(Linescounter==1) // if we have the second line
    {
        k2 = new MinuteLine(clock.this, clockcenter, clock3); // line
for minute
        IndexMinute=3; // mintue on THREE
        r11.addView(k2);
        Checktimer.start(); // timer for checking the answer.
        locks(false); // cannot press any button while checking the
answer.
    }
}
}

```

```

        Linescounter++; // just 2 lines we need - no more and no less ...
    }
}
});
clock4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (Linescounter != 2) {
if(Linescounter==0) // if we have the first line
{
k1 = new ClockLine(clock.this, clockcenter, clock4); // line
for clock
IndexClock=4; // clock on FOUR
if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
{
if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
textToSpeech.speak("NO!! this is not the right
place!!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
r11.removeView(k1); // remove clock line
Linescounter=-1; // we remove the wrong line
}
else
r11.addView(k1); // add clock line
}
else
r11.addView(k1); // add clock line
}
else if(Linescounter==1) // if we have the second line
{
k2 = new MinuteLine(clock.this, clockcenter, clock4); // line
for minute
IndexMinute=4; // mintue on FOUR
r11.addView(k2);
Checktimer.start(); // timer for checking the answer.
locks(false); // cannot press any button while checking the
answer.
}
Linescounter++; // just 2 lines we need - no more and no less ...
}
}
}
});
clock5.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (Linescounter != 2) {
if(Linescounter==0) // if we have the first line
{
k1 = new ClockLine(clock.this, clockcenter, clock5); // line
for clock
IndexClock=5; // clock on FIVE
if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
{
if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
textToSpeech.speak("NO!! this is not the right
place!!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
r11.removeView(k1); // remove clock line
Linescounter=-1; // we remove the wrong line
}
else
r11.addView(k1); // add clock line
}
}
}
}
});

```

```

        else
            r11.addView(k1); // add clock line
        }
        else if(Linescounter==1) // if we have the second line
        {
            k2 = new MinuteLine(clock.this, clockcenter, clock5); // line
    for minute
        IndexMinute=5; // mintue on FIVE
        r11.addView(k2);
        Checktimer.start(); // timer for checking the answer.
        locks(false); // cannot press any button while checking the
    answer.
        }
        Linescounter++; // just 2 lines we need - no more and no less ...
    }
}
});
clock6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2) {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock6); // line
    for clock
                IndexClock=6; // clock on SIX
                if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 1)
                {
                    if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
                        textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                        r11.removeView(k1); // remove clock line
                        Linescounter=-1; // we remove the wrong line
                    }
                    else
                        r11.addView(k1); // add clock line
                }
                else
                    r11.addView(k1); // add clock line
            }
            else if(Linescounter==1) // if we have the second line
            {
                k2 = new MinuteLine(clock.this, clockcenter, clock6); // line
    for minute
                IndexMinute=6; // mintue on SIX
                r11.addView(k2);
                Checktimer.start(); // timer for checking the answer.
                locks(false); // cannot press any button while checking the
    answer.
            }
            Linescounter++; // just 2 lines we need - no more and no less ...
        }
    }
});
clock7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2) {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock7); // line
    for clock

```

```

        IndexClock=7; // clock on SEVEN
        if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
        {
            if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
                textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                rll1.removeView(k1); // remove clock line
                Linescounter=-1; // we remove the wrong line
            }
            else
                rll1.addView(k1); // add clock line
        }
        else
            rll1.addView(k1); // add clock line
    }
    else if(Linescounter==1) // if we have the second line
    {
        k2 = new MinuteLine(clock.this, clockcenter, clock7); // line
for minute
        IndexMinute=7; // mintue on SEVEN
        rll1.addView(k2);
        Checktimer.start(); // timer for checking the answer.
        locks(false); // cannot press any button while checking the
answer.
    }
    Linescounter++; // just 2 lines we need - no more and no less ...
}
}
});
clock8.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    if (Linescounter != 2)
    {
        if(Linescounter==0) // if we have the first line
        {
            k1 = new ClockLine(clock.this, clockcenter, clock8); // line
for clock
            IndexClock=8; // clock on EIGHT
            if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
            {
                if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
                    textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                    rll1.removeView(k1); // remove clock line
                    Linescounter=-1; // we remove the wrong line
                }
                else
                    rll1.addView(k1); // add clock line
            }
            else
                rll1.addView(k1); // add clock line
        }
        else if(Linescounter==1) // if we have the second line
        {
            k2 = new MinuteLine(clock.this, clockcenter, clock8); // line
for minute
            IndexMinute=8; // mintue on EIGHT
            rll1.addView(k2);
            Checktimer.start(); // timer for checking the answer.
            locks(false); // cannot press any button while checking the
        }
    }
}

```

```

answer.

        }
        Linescounter++; // just 2 lines we need - no more and no less ...
    }
}
});

clock9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2)
        {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock9); // line
for clock
                IndexClock=9; // clock on NINE
                if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
                {
                    if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
textToSpeech.speak("NO!! this is not the right
place!!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                        r11.removeView(k1); // remove clock line
                        Linescounter=-1; // we remove the wrong line
                    }
                    else
                        r11.addView(k1); // add clock line
                }
                else
                    r11.addView(k1); // add clock line
            }
            else if(Linescounter==1) // if we have the second line
            {
                k2 = new MinuteLine(clock.this, clockcenter, clock9); // line
for minute
                IndexMinute=9; // mintue on NINE
                r11.addView(k2);
                Checktimer.start(); // timer for checking the answer.
                locks(false); // cannot press any button while checking the
answer.
            }
            Linescounter++; // just 2 lines we need - no more and no less ...
        }
    }
});
clock10.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (Linescounter != 2)
        {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock10); // line
for clock
                IndexClock=10; // clock on TEN
                if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
                {
                    if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
textToSpeech.speak("NO!! this is not the right
place!!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                        r11.removeView(k1); // remove clock line
                        Linescounter=-1; // we remove the wrong line
                }
            }
        }
    }
});

```

```

        }
        else
            r11.addView(k1); // add clock line
    }
    else
        r11.addView(k1); // add clock line
}
else if(Linescounter==1) // if we have the second line
{
    k2 = new MinuteLine(clock.this, clockcenter, clock10); // line
for minute
    IndexMinute=10; // mintue on TEN
    r11.addView(k2);
    Checktimer.start(); // timer for checking the answer.
    locks(false); // cannot press any button while checking the
answer.
}
Linescounter++; // just 2 lines we need - no more and no less ...
}
}
});
clock11.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (Linescounter != 2)
{
    if(Linescounter==0) // if we have the first line
    {
        k1 = new ClockLine(clock.this, clockcenter, clock11); // line
for clock
        IndexClock=11; // clock on ELEVEN
        if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
        {
            if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1){
                textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                r11.removeView(k1); // remove clock line
                Linescounter=-1; // we remove the wrong line
            }
            else
                r11.addView(k1); // add clock line
        }
        else
            r11.addView(k1); // add clock line
    }
    else if(Linescounter==1) // if we have the second line
    {
        k2 = new MinuteLine(clock.this, clockcenter, clock11); // line
for minute
        IndexMinute=11; // mintue on ELEVEN
        r11.addView(k2);
        Checktimer.start(); // timer for checking the answer.
        locks(false); // cannot press any button while checking the
answer.
    }
    Linescounter++; // just 2 lines we need - no more and no less ...
}
}
});
clock12.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

```

```

        if (Linescounter != 2)
        {
            if(Linescounter==0) // if we have the first line
            {
                k1 = new ClockLine(clock.this, clockcenter, clock12); // line
for clock
                IndexClock=0; // clock on TWELVE ( 00:..)
                if(md81.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
                {
                    if(j<8 && IndexClock!=i || j>=8 && IndexClock!=i+1)
                    {
                        textToSpeech.speak("NO!! this is not the right
place!!!", TextToSpeech.QUEUE_ADD, null); // wrong line message
                        r11.removeView(k1); // remove clock line
                        Linescounter=-1; // we remove the wrong line
                    }
                    else
                        r11.addView(k1); // add clock line
                }
                else
                    r11.addView(k1); // add clock line
            }
            else if(Linescounter==1) // if we have the second line
            {
                k2 = new MinuteLine(clock.this, clockcenter, clock12); // line
for minute
                IndexMinute=0; // mintue on TWELVE ( ..:00 )
                r11.addView(k2);
                Checktimer.start(); // timer for checking the answer.
                locks(false); // cannot press any button while checking the
answer.
            }
            Linescounter++; // just 2 lines we need - no more and no less ...
        }
    }
});;
}
// onStop function - for exit we stop the timer and the speech.
@Override
protected void onStop()
{
    super.onStop();
    textToSpeech.shutdown(); // stop speech
    Checktimer.cancel(); // stop timer
    finish(); //to end the Activity immediatly.
}
public class CounterClass extends CountDownTimer
{
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                           to {@link #start()} until the countdown is done
and {@link #onFinish()}
     *
     * @param countDownInterval   The interval along the way to receive
     *                           {@link #onTick(long)} callbacks.
     */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
    @Override
    public void onTick(long millisUntilFinished) {

```

```

        long millis =millisUntilFinished;
        String hms=String.format("%02d:%02d:%02d",
        TimeUnit.MILLISECONDS.toHours(millis),TimeUnit.MILLISECONDS.toMinutes(millis)-
        TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
        TimeUnit.MILLISECONDS.toSeconds(millis)-
        TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis))),;
    }
    // this is the function which run when the time is done !
    @Override
    public void onFinish()
    {
        /* here we have a specific situatoin on the clock - above the 8 it means
you may have now 14:40 or 14:45 or 14:50 or 14:55
        in this situation - we must put the the CLOCK LINE in the next clock
index,according the o'clock rules.
        */
        if (j >= 8)
        {
            if ((i + 1 == 12 && i!=11) || (i==11 && IndexClock==0 && j ==
IndexMinute) || ((i + 1 == IndexClock) && (j == IndexMinute)))
            {
                SCORE++;
                my_db md811 = new my_db(getApplicationContext(), "JAMIL", null,
1); // CALLING DB...

md811.updateScoreClock(Signin.etName1.getText().toString(),SCORE); // update the
score
                // here - you can buy the Clock Helper Feature - if your score >
NEW UPDATED SCORE at the Store Page.
                if (SCORE >= Store.updateclock &&
md811.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 0)
                    Toast.makeText(getApplicationContext(), "you can go to the
Store and buy the Clock Helper Feature!", Toast.LENGTH_SHORT).show();
                    i = rand.nextInt(12); // random line index
                    j = rand.nextInt(12); // random column index
                    currentclock.setText("Write:" + OclockArray[i][j] + "\nScore : "
+ SCORE ); // another clock and score
                }
                else
                {
                    Mp.seekTo(2000); // runs for 2 seconds
                    Mp.start(); // starting
                    currentclock.setText("Write:" + OclockArray[i][j] + "\nScore : "
+ SCORE); // another clock and score
                    r11.startAnimation(AnimationUtils.loadAnimation(clock.this,
R.anim.shaking)); // running the animation ...
                }
            }
            else if (j == IndexMinute && i == IndexClock)
            {
                // there's no exceptions,this is a situation which CLOCK LINE put on
the current clock index
                SCORE++; // one more score
                my_db md812 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
CALLING DB..
                md812.updateScoreClock(Signin.etName1.getText().toString(),SCORE); // //
update the score
                if (SCORE >= Store.updateclock &&
md812.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString()) == 0)
                    Toast.makeText(getApplicationContext(), "you can go to the Store
and buy the Clock Helper Feature!", Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

        i = rand.nextInt(12); // random line index
        j = rand.nextInt(12); // random column index
        currentclock.setText("Write:" + OclockArray[i][j] + "\nScore : " +
SCORE ); // another clock and score
    }
    else
    {
        Mp.seekTo(2000); // runs for 2 seconds
        Mp.start(); // starting
        currentclock.setText("Write:" + OclockArray[i][j] + "\nScore : " +
SCORE ); // another clock and score
        r11.startAnimation(AnimationUtils.loadAnimation(clock.this,
R.anim.shaking)); // running the animation ...

    }
    Linescounter = 0;
    locks(true); // enable buttons
    r11.removeView(k1); // remove clock line // remove the line
    r11.removeView(k2); // remove the line
    Checktimer.cancel();
}
}
// locks function - for lock/unlock clock buttons.
public void locks(boolean bn)
{
    clock1.setEnabled(bn); // bn is true/false
    clock2.setEnabled(bn); // bn is true/false
    clock3.setEnabled(bn); // bn is true/false
    clock4.setEnabled(bn); // bn is true/false
    clock5.setEnabled(bn); // bn is true/false
    clock6.setEnabled(bn); // bn is true/false
    clock7.setEnabled(bn); // bn is true/false
    clock8.setEnabled(bn); // bn is true/false
    clock9.setEnabled(bn); // bn is true/false
    clock10.setEnabled(bn); // bn is true/false
    clock11.setEnabled(bn); // bn is true/false
    clock12.setEnabled(bn); // bn is true/false
}
}

```

קובץ ClockLine.java

מחלקה עוזר למשחק ה-CLOCK

לצייר קו השעה ...

```
package com.example.fares.taguapp;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

/**
 * Created by Jamil-N on 3/30/2016.
 */
public class ClockLine extends View {
    Paint paint = new Paint(); // Paint.
    View startView; // two view - star and end points.
    View endView;

    public ClockLine(Context context, View startView, View endView) {
        super(context);
        paint.setColor(Color.BLACK); // set a color
        paint.setStrokeWidth(6); // set stroke width
        this.startView = startView; // setter startView
        this.endView = endView; // setter endView
    }

    @SuppressLint("NewApi")
    public void onDraw(Canvas canvas) {
        // drawing the line ...
        canvas.drawLine(startView.getX(), startView.getY(),
        ((endView.getX() + startView.getX()) / 2) + 1, ((endView.getY() + startView.getY()) / 2),
        paint);
    }
}
```

קובץ MinuteLine.java

מחלקה עוזר למשחק ה-CLOCK

לצייר קו הדקות ...

```
package com.example.fares.taguapp;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

/**
 * Created by Jamil-N on 3/30/2016.
 */
public class MinuteLine extends View {
    Paint paint = new Paint(); // Paint
    View startView; // two view - start and end points.
    View endView;

    public MinuteLine(Context context, View startView, View endView) {
        super(context);
        paint.setColor(Color.BLACK); // set a color
        paint.setStrokeWidth(4); // set stroke width
        this.startView = startView; // setter
        this.endView = endView; // setter
    }

    @SuppressLint("NewApi")
    public void onDraw(Canvas canvas) {

        canvas.drawLine(startView.getX(), startView.getY(), endView.getX() + 16, endView.getY() + 16,
                paint); // drawing the line ...
    }
}
```

XO.java קובץ

```
package com.example.fares.taguapp;

import android.os.CountDownTimer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Random;
import java.util.concurrent.TimeUnit;

public class XO extends AppCompatActivity {
    Button x,o,c1,c2,c3,c4,c5,c6,c7,c8,c9; // buttons
    boolean flager=false; // flager
    int sog=0,scorex=0,scoreo=0; // sog if X or O, Score for X, Score for O.
    TextView t1; // textview
    final CounterClass timer4 =new CounterClass(2000,1000); // timer variable for 2
seconds.
    int[] arr = new int[10]; // the places of x/o ( 9 nine , 0->9 )
    @Override
    protected void onStop() {
        super.onStop();
        timer4.cancel(); // Stop Timer.
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_xo);
        c1 = (Button)findViewById(R.id.b1); // set button by id ...
        c2 = (Button)findViewById(R.id.b2); // set button by id ...
        c3 = (Button)findViewById(R.id.b3); // set button by id ...
        c4 = (Button)findViewById(R.id.b4); // set button by id ...
        c5 = (Button)findViewById(R.id.b5); // set button by id ...
        c6 = (Button)findViewById(R.id.b6); // set button by id ...
        c7 = (Button)findViewById(R.id.b7); // set button by id ...
        c8 = (Button)findViewById(R.id.b8); // set button by id ...
        c9 = (Button)findViewById(R.id.b9); // set button by id ...
        x = (Button)findViewById(R.id.buttonX); // set button by id ...
        o = (Button)findViewById(R.id.buttonO); // set button by id ...
        t1 = (TextView)findViewById(R.id.chooseXO); // set textview by id ...
        my_listener m1 = new my_listener(); // we have m1 variable of my_listener
class that helps on OnClickListerner events on every button.
        x.setOnClickListener(m1); // connect button to the class
        o.setOnClickListener(m1); // connect button to the class
        c1.setOnClickListener(m1); // connect button to the class
        c2.setOnClickListener(m1); // connect button to the class
        c3.setOnClickListener(m1); // connect button to the class
        c4.setOnClickListener(m1); // connect button to the class
        c5.setOnClickListener(m1); // connect button to the class
        c6.setOnClickListener(m1); // connect button to the class
        c7.setOnClickListener(m1); // connect button to the class
        c8.setOnClickListener(m1); // connect button to the class
        c9.setOnClickListener(m1); // connect button to the class
    }

    class my_listener implements View.OnClickListener {
```

```

@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.buttonX :
            sog = 1; // 1 means we chose X
            x.setVisibility(View.GONE); // X Option button disappeared
            o.setVisibility(View.GONE); // O Option button disappeared
            t1.setVisibility(View.GONE); // text "choose X or O" disappeared
            StartWork(); // StartWork() Function - to reset the places of
X/O, the array of nine places, everything for the next match.
            break;
        case R.id.buttonO :
            x.setVisibility(View.GONE); // X Option button disappeared
            o.setVisibility(View.GONE); // O Option button disappeared
            t1.setVisibility(View.GONE); // text "choose X or O" disappeared
            sog = 2; // 2 means we chose O
            StartWork(); // StartWork() Function - to reset the places of
X/O, the array of nine places, everything for the next match.
            break;
        case R.id.b1 :
            if(check(1)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :
            {
                flager=true; // flager ON
                setplace(sog,1,c1);
            }
            break;
        case R.id.b2 :
            if(check(2)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :
            {
                flager=true;
                setplace(sog,2,c2);
            }
            break;
        case R.id.b3 :
            if(check(3)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :
            {
                flager=true;
                setplace(sog,3,c3);
            }
            break;
        case R.id.b4 :
            if(check(4)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :
            {
                flager=true;
                setplace(sog,4,c4);
            }
            break;
        case R.id.b5 :
            if(check(5)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :
            {
                flager=true;
                setplace(sog,5,c5);
            }
            break;
        case R.id.b6 :
            if(check(6)&&sog!=0) // check function is checking if this place
is empty of not - if its empty then :

```

```

        {
            flager=true;
            setplace(sog,6,c6);
        }
        break;
    case R.id.b7 :
        if(check(7)&&sog!=0) // check function is checking if this place
is empty or not - if its empty then :
{
            flager=true;
            setplace(sog,7,c7);
}
        break;
    case R.id.b8 :
        if(check(8)&&sog!=0) // check function is checking if this place
is empty or not - if its empty then :
{
            flager=true;
            setplace(sog,8,c8);
}
        break;
    case R.id.b9 :
        if(check(9)&&sog!=0) // check function is checking if this place
is empty or not - if its empty then :
{
            flager=true;
            setplace(sog,9,c9);
}
        break;
}
}
private boolean check(int index) // check() Fucntion - to check if the "index"
place is empty or not - True if empty, False if not empty.
{
    if(arr[index-1]==0)
        return true; // empty.
    else
        return false; // not empty.
}

private void setplace(int sog , int index,Button c) // setplace() Function - to
set X or O on a specific place ( index ) and specific Button ( c )
{
    arr[index-1]=sog; // because of this place is empty then we put in this place
the sog ( or X or O )
    if(sog==2) // if its O - we put O Image
        c.setBackgroundResource(R.drawable.o);
    else // if its X - we put X Image
        c.setBackgroundResource(R.drawable.xxw);
    checker(); // checker() is running ...
}
private void checker() // after set place ( X or O ) we check what we have now.
{
    // 1.Situation that X wins OR 2.Situation that O Wins OR 3.no one win yet and
in this situation we call setrandom() Fucntion.
    // X has 8 options to win - every place in the array = place for X or O.
    if((arr[0]== arr[1] && arr[0] == arr[2] && arr[0]==1) ||
       (arr[3]==arr[4]&&arr[3]==arr[5] && arr[3]==1) ||
       (arr[6]==arr[7]&&arr[6]==arr[8] && arr[8]==1) ||
       (arr[0]==arr[3]&&arr[0]==arr[6] && arr[0]==1) ||
       (arr[1]==arr[4]&&arr[1]==arr[7] && arr[1]==1) ||
       (arr[2]==arr[5]&&arr[2]==arr[8] && arr[2]==1) ||
       (arr[0]==arr[1]&&arr[0]==arr[2] && arr[0]==1) ||
       (arr[3]==arr[4]&&arr[3]==arr[5] && arr[3]==1) ||
       (arr[6]==arr[7]&&arr[6]==arr[8] && arr[6]==1) ||
       (arr[0]==arr[4]&&arr[0]==arr[5] && arr[0]==1) ||
       (arr[1]==arr[5]&&arr[1]==arr[6] && arr[1]==1) ||
       (arr[2]==arr[3]&&arr[2]==arr[6] && arr[2]==1) ||
       (arr[4]==arr[5]&&arr[4]==arr[6] && arr[4]==1) ||
       (arr[7]==arr[8]&&arr[7]==arr[8] && arr[7]==1))
}

```

```

        (arr[0]==arr[4]&&arr[0]==arr[8] && arr[0]==1) ||
        (arr[2]==arr[4]&&arr[2]==arr[6] && arr[2]==1))
    {
        scorex++; // one more score for X ...
        Toast.makeText(getApplicationContext(),"x score : "+ scorex + "\no score :
"+scoreo,Toast.LENGTH_LONG).show(); // X & O Scores.
        locks(false); // when application resets everything - we cant click any
button.
        timer4.start(); // timer to reset and get ready and play again.
    }
    else // O has 8 options to win - every place in the array = place for X or O.
    {
        if((arr[0]== arr[1] && arr[0] == arr[2] && arr[0]==2) ||
           (arr[3]==arr[4]&&arr[3]==arr[5] && arr[3]==2) ||
           (arr[6]==arr[7]&&arr[6]==arr[8] && arr[8]==2) ||
           (arr[0]==arr[3]&&arr[0]==arr[6] && arr[0]==2) ||
           (arr[1]==arr[4]&&arr[1]==arr[7] && arr[1]==2) ||
           (arr[2]==arr[5]&&arr[2]==arr[8] && arr[2]==2) ||
           (arr[0]==arr[4]&&arr[0]==arr[8] && arr[0]==2) ||
           (arr[2]==arr[4]&&arr[2]==arr[6] && arr[2]==2))
        {
            scoreo++; // one more score for O ...
            Toast.makeText(getApplicationContext(),"x score : "+ scorex + "\no
score : "+scoreo,Toast.LENGTH_LONG).show(); // X & O Scores.
            locks(false); // when application resets everything - we cant click
any button.
            timer4.start(); // timer to reset and get ready and play again.
        }
        else if(flager==true) // if any player ( X or O ) does NOT win yet ...
CALLING setrandom() Function ...
        setrandom(); // CALLING setrandom() Function ...
    }

private void setrandom()
{
    flager = false;
    int rand=1,counter2=0,j;
    int sogComp; // computer player.
    Random i = new Random(); // rand variable.
    if(sog==1) // if its X
        sogComp=2; // the computer O
    else
        sogComp=1; // if its O -> the computer X
    for(j=0;j<9;j++) // loop
    {
        if(arr[j]==0)
            counter2++; // counting how many places are empty every round.
    }
    if(counter2>0) // if there's an empty places ...
    {
        // here we have 24 options to check - here the computer is looking for a
situation for him to win.
        // we have 8 options to win : 3 columns or 3 lines or the two diagonals
are the same ( X or O ) So :
        // every option checks which place is empty of their three places.
        if (arr[0] == arr[1] && arr[2] == 0 && arr[0] == sogComp) {
            arr[2] = sogComp;
            rand = 2;
        } else if (arr[1] == arr[2] && arr[0] == 0 && arr[1] == sogComp) {
            arr[0] = sogComp;
            rand = 0;
        } else if (arr[2] == arr[0] && arr[1] == 0 && arr[2] == sogComp) {
            rand = 1;
            arr[1] = sogComp;
        }
    }
}

```

```

} else if (arr[3] == arr[4] && arr[5] == 0 && arr[3]== sogComp) {
    arr[5] = sogComp;
    rand = 5;
} else if (arr[3] == arr[5] && arr[4] == 0 && arr[3] == sogComp) {
    rand = 4;
    arr[4] = sogComp;
} else if (arr[4] == arr[5] && arr[3] == 0 && arr[4] == sogComp) {
    rand = 3;
    arr[3] = sogComp;
} else if (arr[6] == arr[7] && arr[8] == 0 && arr[6] == sogComp) {
    rand = 8;
    arr[8] = sogComp;
} else if (arr[6] == arr[8] && arr[7] == 0 && arr[6] == sogComp) {
    rand = 7;
    arr[7] = sogComp;
} else if (arr[7] == arr[8] && arr[6] == 0 && arr[7] == sogComp) {
    rand = 6;
    arr[6] = sogComp;
} else if (arr[0] == arr[3] && arr[6] == 0 && arr[0] == sogComp) {
    arr[6] = sogComp;
    rand = 6;
} else if (arr[0] == arr[6] && arr[3] == 0 && arr[0] == sogComp) {
    arr[3] = sogComp;
    rand = 3;
} else if (arr[6] == arr[3] && arr[0] == 0 && arr[6] == sogComp) {
    rand = 0;
    arr[0] = sogComp;
} else if (arr[1] == arr[4] && arr[7] == 0 && arr[1] == sogComp) {
    arr[7] = sogComp;
    rand = 7;
} else if (arr[1] == arr[7] && arr[4] == 0 && arr[1] == sogComp) {
    arr[4] = sogComp;
    rand = 4;
} else if (arr[4] == arr[7] && arr[1] == 0 && arr[4] == sogComp) {
    arr[1] = sogComp;
    rand = 1;
} else if (arr[2] == arr[5] && arr[8] == 0 && arr[2] == sogComp) {
    arr[8] = sogComp;
    rand = 8;
} else if (arr[2] == arr[8] && arr[5] == 0 && arr[2] == sogComp) {
    arr[5] = sogComp;
    rand = 5;
} else if (arr[5] == arr[8] && arr[2] == 0 && arr[5] == sogComp) {
    arr[2] = sogComp;
    rand = 2;
} else if (arr[0] == arr[4] && arr[8] == 0 && arr[0] == sogComp) {
    arr[8] = sogComp;
    rand = 8;
} else if (arr[0] == arr[8] && arr[4] == 0 && arr[0] == sogComp) {
    arr[4] = sogComp;
    rand = 4;
} else if (arr[8] == arr[4] && arr[0] == 0 && arr[8] == sogComp) {
    arr[0] = sogComp;
    rand = 0;
} else if (arr[2] == arr[4] && arr[6] == 0 && arr[2] == sogComp) {
    arr[6] = sogComp;
    rand = 6;
} else if (arr[2] == arr[6] && arr[4] == 0 && arr[2] == sogComp) {
    arr[4] = sogComp;
    rand = 4;
} else if (arr[6] == arr[4] && arr[2] == 0 && arr[6] == sogComp) {
    arr[2] = sogComp;
    rand = 2;
}

```

```

    } // now here we have another 24 options to check - here the computer is
    // trying to prevent the user to win.
    // we have 8 options to win : 3 columns or 3 lines or the two diagonals
    // are the same ( X or O ) So :
    // every option checks which place is empty of their three places.
    else if (arr[0] == arr[1] && arr[2] == 0 && arr[0] != 0) {
        arr[2] = sogComp;
        rand = 2;
    } else if (arr[1] == arr[2] && arr[0] == 0 && arr[1] != 0) {
        arr[0] = sogComp;
        rand = 0;
    } else if (arr[2] == arr[0] && arr[1] == 0 && arr[2] != 0) {
        rand = 1;
        arr[1] = sogComp;
    } else if (arr[3] == arr[4] && arr[5] == 0 && arr[3] != 0) {
        arr[5] = sogComp;
        rand = 5;
    } else if (arr[3] == arr[5] && arr[4] == 0 && arr[3] != 0) {
        rand = 4;
        arr[4] = sogComp;
    } else if (arr[4] == arr[5] && arr[3] == 0 && arr[4] != 0) {
        rand = 3;
        arr[3] = sogComp;
    } else if (arr[6] == arr[7] && arr[8] == 0 && arr[6] != 0) {
        rand = 8;
        arr[8] = sogComp;
    } else if (arr[6] == arr[8] && arr[7] == 0 && arr[6] != 0) {
        rand = 7;
        arr[7] = sogComp;
    } else if (arr[7] == arr[8] && arr[6] == 0 && arr[7] != 0) {
        rand = 6;
        arr[6] = sogComp;
    } else if (arr[0] == arr[3] && arr[6] == 0 && arr[0] != 0) {
        arr[6] = sogComp;
        rand = 6;
    } else if (arr[0] == arr[6] && arr[3] == 0 && arr[0] != 0) {
        arr[3] = sogComp;
        rand = 3;
    } else if (arr[6] == arr[3] && arr[0] == 0 && arr[6] != 0) {
        rand = 0;
        arr[0] = sogComp;
    } else if (arr[1] == arr[4] && arr[7] == 0 && arr[1] != 0) {
        arr[7] = sogComp;
        rand = 7;
    } else if (arr[1] == arr[7] && arr[4] == 0 && arr[1] != 0) {
        rand = 4;
        arr[4] = sogComp;
    } else if (arr[4] == arr[7] && arr[1] == 0 && arr[4] != 0) {
        rand = 1;
        arr[1] = sogComp;
    } else if (arr[2] == arr[5] && arr[8] == 0 && arr[2] != 0) {
        arr[8] = sogComp;
        rand = 8;
    } else if (arr[2] == arr[8] && arr[5] == 0 && arr[2] != 0) {
        rand = 5;
        arr[5] = sogComp;
    } else if (arr[5] == arr[8] && arr[2] == 0 && arr[5] != 0) {
        rand = 2;
        arr[2] = sogComp;
    } else if (arr[0] == arr[4] && arr[8] == 0 && arr[0] != 0) {
        arr[8] = sogComp;
        rand = 8;
    } else if (arr[0] == arr[8] && arr[4] == 0 && arr[0] != 0) {

```

```

        arr[4] = sogComp;
        rand = 4;
    } else if (arr[8] == arr[4] && arr[0] == 0 && arr[8] != 0) {
        arr[0] = sogComp;
        rand = 0;
    } else if (arr[2] == arr[4] && arr[6] == 0 && arr[2] != 0) {
        arr[6] = sogComp;
        rand = 6;
    } else if (arr[2] == arr[6] && arr[4] == 0 && arr[2] != 0) {
        arr[4] = sogComp;
        rand = 4;
    } else if (arr[6] == arr[4] && arr[2] == 0 && arr[6] != 0) {
        arr[2] = sogComp;
        rand = 2;
    }
    else // if there is no situation for the computer to win so the computer
puts his X or his O randomly in empty place.
{
    do {
        rand = i.nextInt(9);
    } while (arr[rand] != 0); // while its not empty ...
    if (sog == 2) // if you are O
        arr[rand] = 1; // the computer puts X
    else
        arr[rand] = 2; // if you are X then the computer puts O
}
switch (rand) // according to the empty place.
{
    case 0:
        if(sog==2) // if its X put X else O
            c1.setBackgroundResource(R.drawable.xxw); // X image
        else
            c1.setBackgroundResource(R.drawable.o); // O image
        break;
    case 1:
        if(sog==2) // if its X put X else O
            c2.setBackgroundResource(R.drawable.xxw); // X image
        else
            c2.setBackgroundResource(R.drawable.o); // O image
        break;
    case 2:
        if(sog==2) // if its X put X else O
            c3.setBackgroundResource(R.drawable.xxw); // X image
        else
            c3.setBackgroundResource(R.drawable.o); // O image
        break;
    case 3:
        if(sog==2) // if its X put X else O
            c4.setBackgroundResource(R.drawable.xxw); // X image
        else
            c4.setBackgroundResource(R.drawable.o); // O image
        break;
    case 4:
        if(sog==2) // if its X put X else O
            c5.setBackgroundResource(R.drawable.xxw); // X image
        else
            c5.setBackgroundResource(R.drawable.o); // O image
        break;
    case 5:
        if(sog==2) // if its X put X else O
            c6.setBackgroundResource(R.drawable.xxw); // X image
        else
            c6.setBackgroundResource(R.drawable.o); // O image
}

```

```

        break;
    case 6:
        if(sog==2) // if its X put X else O
            c7.setBackgroundResource(R.drawable.xxw); // X image
        else
            c7.setBackgroundResource(R.drawable.o); // O image
        break;
    case 7:
        if(sog==2) // if its X put X else O
            c8.setBackgroundResource(R.drawable.xxw); // X image
        else
            c8.setBackgroundResource(R.drawable.o); // O image
        break;
    case 8:
        if(sog==2) // if its X put X else O
            c9.setBackgroundResource(R.drawable.xxw); // X image
        else
            c9.setBackgroundResource(R.drawable.o); // O image
        break;
    }
    checker(); // after set place ( X or O ) we check what we have now.
}
else // if counter2=0 it means that count of empty places is ZERO it means ->
draw - we set all the places and no one win.
{
    Toast.makeText(getApplicationContext(), "draw", Toast.LENGTH_LONG).show();
// draw message.
locks(false); // when application resets everything - we cant click any
button.
StartWork(); // StartWork() Fucntion - to reset the places of X/O,the
array of nine places,everything for the next match.
timer4.start(); // timer to reset and get ready and play again.
}
}

// locks function - lock/unlock buttons.
public void locks(boolean lock)
{
    // lock variable can be OR true OR false ...
    c1.setEnabled(lock);
    c2.setEnabled(lock);
    c3.setEnabled(lock);
    c4.setEnabled(lock);
    c5.setEnabled(lock);
    c6.setEnabled(lock);
    c7.setEnabled(lock);
    c8.setEnabled(lock);
    c9.setEnabled(lock);
}
public class CounterClass extends CountDownTimer
{
    /**
     * @param millisInFuture      The number of millis in the future from the call
     *                            to {@link #start()} until the countdown is done
     and {@link #onFinish()} is called.
     * @param countDownInterval   The interval along the way to receive
     *                            {@link #onTick(long)} callbacks.
     */
    public CounterClass(long millisInFuture, long countDownInterval) {
        super(millisInFuture, countDownInterval);
    }
    //here the time is running..
}

```

```

@Override
public void onTick(long millisUntilFinished) {
    long millis =millisUntilFinished;
    String hms=String.format("%02d:%02d:%02d",
TimeUnit.MILLISECONDS.toHours(millis),TimeUnit.MILLISECONDS.toMinutes(millis)-
TimeUnit.HOURS.toMinutes(TimeUnit.MILLISECONDS.toHours(millis)),
TimeUnit.MILLISECONDS.toSeconds(millis)-
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(millis)));
}
// this is the function which run when the time is done !
@Override
public void onFinish() {
    locks(true); // turn ON - we can click the buttons.
    StartWork(); // StartWork() Fucntion - to reset the places of X/O,the
array of nine places,everything for the next match.
}
}
private void StartWork() // StartWork() Fucntion - to reset the places of X/O,the
array of nine places,everything for the next match.
{
    flager=false; // set flager FALSE
    c1.setBackgroundResource(R.drawable.t111); // empty photo
    c2.setBackgroundResource(R.drawable.t111); // empty photo
    c3.setBackgroundResource(R.drawable.t111); // empty photo
    c4.setBackgroundResource(R.drawable.t111); // empty photo
    c5.setBackgroundResource(R.drawable.t111); // empty photo
    c6.setBackgroundResource(R.drawable.t111); // empty photo
    c7.setBackgroundResource(R.drawable.t111); // empty photo
    c8.setBackgroundResource(R.drawable.t111); // empty photo
    c9.setBackgroundResource(R.drawable.t111); // empty photo
    for(int i=0;i<9;i++)
        arr[i]=0; // RESET!
}
}

```

קובץ Store.java

```
package com.example.fares.taguapp;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Store extends AppCompatActivity {

    static Button
buyType,buyPhotos,buyMath,buyClock,buyColors,typescore,photoscore,mathscore,clockscore
,colorsscore; // buttons
    Button priceMath,priceTyping,pricePhotos,priceClock,priceColors; // buttons
    static int
flagType=0,flagPhotos=0,flagMath=0,flagClock=0,flagColors=0,updatetyping=2,updatephoto
s=30,updatemath=5,updateclock=10,updatecolors=120;
    int scoreType,scorePhotos,scoreMath,scoreClock,scoreColors; // scores variables.
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_store);
        // button declarations ...
        priceMath=(Button)findViewById(R.id.pricemath); // button declarations ...
        pricePhotos=(Button)findViewById(R.id.pricephotos); // button declarations ...
        priceTyping=(Button)findViewById(R.id.typeprice); // button declarations ...
        priceClock=(Button)findViewById(R.id.priceclock); // button declarations ...
        priceColors=(Button)findViewById(R.id.colorprice); // button declarations ...
        buyType=(Button)findViewById(R.id.btnTyping); // button declarations ...
        buyPhotos=(Button)findViewById(R.id.btnPhotos); // button declarations ...
        buyMath=(Button)findViewById(R.id.btnMath); // button declarations ...
        buyClock=(Button)findViewById(R.id.btnClock); // button declarations ...
        buyColors=(Button)findViewById(R.id.btnColors); // button declarations ...
        typescore=(Button)findViewById(R.id.typescore); // button declarations ...
        photoscore=(Button)findViewById(R.id.photoscore); // button declarations ...
        mathscore=(Button)findViewById(R.id.mathscore); // button declarations ...
        clockscore=(Button)findViewById(R.id.clockscore); // button declarations ...
        colorsscore=(Button)findViewById(R.id.colorscore); // button declarations ...
        priceMath.setText(Integer.toString(updatemath)); // price math text
        pricePhotos.setText(Integer.toString(updatephotos)); // price photos text
        // ****
        final my_db md200 = new my_db(getApplicationContext(), "JAMIL", null, 1); // //
CALLING DATABASE ...
        // we get the last update scores - Type & Photos & Math & Clock & Colors
    SCORES ->
        scoreType=md200.getScoreType(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        scorePhotos=md200.getScorePhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        scoreMath=md200.getScoreMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        scoreClock=md200.getScoreClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        scoreColors=md200.getScoreColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        // store updating score colors ...
        updatecolors=md200.getStoreColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
        if(md200.getStoreFlagColors(Signin.etName1.getText().toString(),

```

```

Signin.etPass1.getText().toString() ==0)
{
    updatecolors=20; // set new colors score to next feature ...
    priceColors.setText(Integer.toString(updatecolors));
    if(scoreColors<updatecolors)
        buyColors.setEnabled(false); // "get feature" button - OFF
    else
        buyColors.setEnabled(true); // "get feature" button - ON
}
else if(md200.getStoreflagColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() ==1)
{
    updatecolors=40; // set new colors score to next feature ...
    priceColors.setText(Integer.toString(updatecolors)); // new score
    if(scoreColors<updatecolors)
        buyColors.setEnabled(false); // "get feature" button - OFF
    else
        buyColors.setEnabled(true); // "get feature" button - ON
}
else if(md200.getStoreflagColors(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() ==2)
{
    priceColors.setText("DONE"); // done - no more features ...
    buyColors.setEnabled(false); // "get feature" button - OFF
}

//*****
***** // store updating score clock ...
updateclock=md200.getStoreClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
if(md200.getStoreflagClock(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() ==1)
{
    priceClock.setText("DONE"); // no more features
    buyClock.setEnabled(false); // "get feature" button - OFF
}
else
{
    updateclock=5; // one feature ...
    if(scoreClock<updateclock)
        buyClock.setEnabled(false); // "get feature" button - OFF
    else
        buyClock.setEnabled(true); // "get feature" button - ON
    priceClock.setText(Integer.toString(updateclock)); // clock price text
...
}

//*****
***** // store updating score typing ...
updatetyping=md200.getStoreTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
if(md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString() ==0)
{
    updatetyping=2; // first feature ...
    priceTyping.setText(Integer.toString(updatetyping)); // typing price text
...
    if(scoreType<updatetyping)
        buyType.setEnabled(false); // "get feature" button - OFF
    else
        buyType.setEnabled(true); // "get feature" button - ON
}

```

```

        }
        else if(md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
        {
            updatetyping=7; // second feature
            priceTyping.setText(Integer.toString(updatetyping)); // typing price text
...
            if(scoreType<updatetyping)
                buyType.setEnabled(false); // "get feature" button - OFF
            else
                buyType.setEnabled(true); // "get feature" button - ON
        }
        else if(md200.getStoreflagTyping(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
        {
            priceTyping.setText("DONE"); // no more features ...
            buyType.setEnabled(false); // "get feature" button - OFF
        }

//*****
***** // store updating score math ...
updateMath=md200.getStoreMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
if(md200.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==0)
{
    updateMath=5; // first feature
    priceMath.setText(Integer.toString(updateMath)); // math price text
    if(scoreMath<updateMath)
        buyMath.setEnabled(false); // "get feature" button - OFF
    else
        buyMath.setEnabled(true); // "get feature" button - ON
}
else if(md200.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
{
    updateMath=10;
    priceMath.setText(Integer.toString(updateMath)); // math price text
    if(scoreMath<updateMath)
        buyMath.setEnabled(false); // "get feature" button - OFF
    else
        buyMath.setEnabled(true); // "get feature" button - ON
}
else if(md200.getStoreflagMath(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
{
    priceMath.setText("DONE"); // no more features ...
    buyMath.setEnabled(false); // "get feature" button - OFF
}

//*****
***** // store updating score photos ...
updatePhotos=md200.getStorePhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString());
if(md200.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==0)
{
    updatePhotos=10; // first feature ...
    pricePhotos.setText(Integer.toString(updatePhotos)); // photos price text
...
    if(scorePhotos<updatePhotos)

```

```

        buyPhotos.setEnabled(false); // "get feature" button - OFF
    else
        buyPhotos.setEnabled(true); // "get feature" button - ON
    }
    else if(md200.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==1)
    {
        updatephotos=15; // second feature ...
        pricePhotos.setText(Integer.toString(updatephotos)); // photos price text
    ...
        if(scorePhotos<updatephotos)
            buyPhotos.setEnabled(false); // "get feature" button - OFF
        else
            buyPhotos.setEnabled(true); // "get feature" button - ON
    }
    else if(md200.getStoreflagPhotos(Signin.etName1.getText().toString(),
Signin.etPass1.getText().toString())==2)
    {
        pricePhotos.setText("DONE"); // no more features ...
        buyPhotos.setEnabled(false); // "get feature" button - OFF
    }
    // show scores ->
    typescore.setText(" "+scoreType+" ");
    photoscore.setText(" "+scorePhotos+" ");
    clockscore.setText(" "+ scoreClock + " ");
    colorsscore.setText(" "+scoreColors+" ");
    mathscore.setText(" "+scoreMath+" ");

//*****
buyType.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(flagType==0)
            flagType = 1; // to next feature ...
            updatotyping=7;
            priceTyping.setText(Integer.toString(updatotyping)); // typing
price text
            md200.StoreflagTypingSave(Signin.etName1.getText().toString(),
flagType); // updating status for features in DB.
            buyType.setEnabled(false); // typing OFF
            if(scoreType>=updatotyping)
                buyType.setEnabled(true); // typing ON
        }
        else if(flagType==1) {
            flagType = 2;
            md200.StoreflagTypingSave(Signin.etName1.getText().toString(),
flagType); // updating status for features in DB.
            priceTyping.setText("DONE"); // no more features
            buyType.setEnabled(false); // typing OFF
        }
    }
});;
buyPhotos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (flagPhotos == 0) {
            flagPhotos = 1; // to next feature ...
            md200.StoreflagPhotosSave(Signin.etName1.getText().toString(),
flagPhotos); // updating status for features in DB.
            buyPhotos.setEnabled(false); // photos OFF
            updatephotos = 15;
            if(scorePhotos>=updatephotos)
                buyPhotos.setEnabled(true); // photos ON
    }
}
);

```

```

        pricePhotos.setText(Integer.toString(updatephotos)); // photos
price text
    } else if (flagPhotos == 1) {
        flagPhotos = 2; // done
        md200.StoreflagPhotosSave(Signin.etName1.getText().toString(),
flagPhotos); // updating status for features in DB.
        buyPhotos.setEnabled(false);
        pricePhotos.setText("DONE!"); // no more features ...
    }
}
});
buyMath.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (flagMath == 0) {
            flagMath = 1; // to next feature ...
            md200.StoreflagMathSave(Signin.etName1.getText().toString(),
flagMath); // updating status for features in DB.
            buyMath.setEnabled(false); // math OFF
            updatemath = 10;
            if(scoreMath>=updatemath)
                buyMath.setEnabled(true); // math ON
            priceMath.setText(Integer.toString(updatemath)); // math price
text
        } else if (flagMath == 1) {
            flagMath = 2; // done
            md200.StoreflagMathSave(Signin.etName1.getText().toString(),
flagMath); // updating status for features in DB.
            buyMath.setEnabled(false);
            updatemath=15;
            priceMath.setText("DONE!"); // no more features ...
        }
    }
});
buyClock.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(flagClock==0) {
            flagClock = 1;
            md200.StoreflagClockSave(Signin.etName1.getText().toString(),
flagClock); // updating status for features in DB.
            buyClock.setEnabled(false); // clock OFF
            priceClock.setText("Done!"); // no more features ...
        }
    }
});
buyColors.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (flagColors == 0) {
            flagColors = 1; // to next feature ...
            md200.StoreflagColorsSave(Signin.etName1.getText().toString(),
flagColors); // updating status for features in DB.
            buyColors.setEnabled(false);
            updatecolors = 40;
            if(scoreColors>=updatecolors)
                buyColors.setEnabled(true); // colors ON
            priceColors.setText(Integer.toString(updatecolors)); // colors
price text
        } else if (flagColors == 1) {
            flagColors = 2; // done
            md200.StoreflagColorsSave(Signin.etName1.getText().toString(),
flagColors); // updating status for features in DB.
        }
    }
});

```

```
        buyColors.setEnabled(false); // colors OFF
        updatecolors = 60;
        priceColors.setText("DONE!");
    }
}
});
```

קובץ UserInfo.java

```
package com.example.fares.taguapp;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class UserInfo extends AppCompatActivity {
    Button ChangePassword; // buttons
    TextView YourName, YourEmail, YourPassword, OurScore; // textview
    String user, pass,str1 = "", str2 = ""; // string
    int flagletter, flagnumbers; // integer
    // PasswordChanger Function - to change your password inside the system - NOT at
    // the sign in page.
    public void PasswordChanger() // PasswordChanger Function - to change your
    password - inside the game.
    {
        AlertDialog.Builder b = new AlertDialog.Builder(UserInfo.this); // Alert
        Dialog ...
        final AlertDialog.Builder b2 = new AlertDialog.Builder(UserInfo.this); // Another Alert Dialog ...
        b.setTitle("Please enter new password :"); // title : enter new password
        b2.setTitle("Please confirm new password :"); // title : confirm new password
        final EditText input = new EditText(UserInfo.this); // final to variables to
        make its values not to be changed.
        final EditText input2 = new EditText(UserInfo.this); // final to variables to
        make its values not to be changed.
        b.setView(input);
        b.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int whichButton) {
                // here we are checking the password according to the password RULES.
                /*
                    1. Minimum of 8 characters in length.
                    2. Password is FAIR if the length = 8.
                    3. Password is WEAK if Password LENGTH > 8 AND Password JUST includes
                    [a-z]/[A-Z] WITHOUT [0-9]
                    4. Password is STRONG if Password LENGTH > 8 AND Password includes [a-
                    z]/[A-Z] AND [0-9]
                */
                str1 = input.getText().toString(); // get the FIRST string.
                String Passing = input.getText().toString();
                for (int indexer0 = 0; indexer0 <= input.length() - 1;
                indexer0++)
                {
                    if ((Passing.charAt(indexer0) >= 'a' &&
                    Passing.charAt(indexer0) <= 'z') || (Passing.charAt(indexer0) >= 'A' &&
                    Passing.charAt(indexer0) <= 'Z'))
                        flagletter = 1; // if the string has letters.
                    if (Passing.charAt(indexer0) >= '0' &&
                    Passing.charAt(indexer0) <= '9')
```

```

                flagnumbers = 1; // if the string has numbers.
            }
            if (input.length() < 8)
            {
                Toast.makeText(getApplicationContext(), "Minimum of 8
characters in length", Toast.LENGTH_SHORT).show();
                PasswordChanger(); // PasswordChanger() Function ->
Recursion : do it again if you still wanna change your password.
            }
            else
            {
                if (input.length() == 8) // if len = 8...
                    Toast.makeText(getApplicationContext(), "FAIR",
Toast.LENGTH_SHORT).show(); // fair pass
                else if (input.length() > 8 && flagnumbers == 0 &&
flagletter == 1)
                {
                    flagletter = 0;
                    Toast.makeText(getApplicationContext(), "WEAK",
Toast.LENGTH_SHORT).show(); // weak pass
                }
                else if (input.length() > 8 && flagnumbers == 1 &&
flagletter == 1)
                {
                    flagletter = flagnumbers = 0;
                    Toast.makeText(getApplicationContext(), "Strong",
Toast.LENGTH_SHORT).show(); // strong pass
                }
                flagletter = flagnumbers = 0; // reseting the helper
varilabes for the next time.
            }
            b2.setView(input2);
            b2.setPositiveButton("OK", new
DialogInterface.OnClickListener()
{
    @Override
    public void onClick(DialogInterface dialog,
int whichButton)
    {
        // here we must confirm the new password...
        str2 = input2.getText().toString(); // get
the SECOND password ...
        if (str1.equals(str2)) { // checking if
the password is the same AGAIN.
            my_db.passwordchanger = new
my_db(getApplicationContext(), "JAMIL", null, 1); // CALLING DB ...
            YourPassword.setText("Password : " +
str1); // new password
            Signin.etPass1.setText(str1); // we
save the new one in Signin Page.

passwordchanger.updateCurrentPassword(Signin.etName1.getText().toString(), str1); // updating new pass ...

Toast.makeText(getApplicationContext(), "Password has been changed!",
Toast.LENGTH_LONG).show();
    }
    else
    {
        str1 = str2 = ""; // reset and "not
the same password" message.

Toast.makeText(getApplicationContext(), "Not The Same Password!",
Toast.LENGTH_SHORT).show();
        PasswordChanger(); // Recursion - do
it again if you still wanna change your password.
    }
}

```


קובץ About.java

```
package com.example.fares.taguapp;

import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.TextView;
import android.widget.VideoView;

public class About extends AppCompatActivity {
    TextView txtAbout1;
    static TextView txtAbout2;
    VideoView mVideoView;
    Button btn,btn2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
        mVideoView = (VideoView) findViewById(R.id.videoView);
        btn=(Button) findViewById(R.id.playvideo);
        btn2=(Button) findViewById(R.id.pausevideo);
        String uriPath = "android.resource://" + getPackageName() + "/" + R.raw.tagu;
        Uri uri = Uri.parse(uriPath);
        mVideoView.setVideoURI(uri);
        mVideoView.requestFocus();
        mVideoView.start();
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mVideoView.start();
            }
        });
        btn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mVideoView.pause();
            }
        });
    }
}
```

קובץ my_db.java

```
package com.example.fares.taguapp;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import android.widget.Toast;

public class my_db extends SQLiteOpenHelper {
    public my_db(Context context, String name, SQLiteDatabase.CursorFactory factory,
    int version) {
        super(context, name, factory, version);
    }
    // creating the table.
    @Override
    public void onCreate(SQLiteDatabase db)
    {
        String query;
        query="create table if not exists player(name text,pass text,mScore
integer,letter integer,typing integer,clock integer,colors integer,photos
integer,focus integer,storeClock integer,flagClock integer,storeTyping
integer,flagTyping integer,storeColors integer,flagColors integer,storeMath
integer,flagMath integer,storePhotos integer,flagPhotos integer,mail text,cube
integer,cubetwo integer,arrlist text)";
        db.execSQL(query);
    }

    // Adding a new user for the table.
    public void add_player(String name , String pass,String mail) // add player and
set score zero to the math game.
    {
        SQLiteDatabase db=getWritableDatabase();
        String query; // by insert action ... set score ZERO to math game.
        query="insert into player(mail,name,pass,mScore) values
('"+mail+"','"+name+"','"+pass+"','"+0+"')";
        Log.i("q1", query);
        db.execSQL(query);
    }
    /*when the user sign in - we are checking if this is the first time the user sign
in so we tell him "Welcome,User!"
    but if this is not the first time for the user sign in so we tell him "Welcome
Back,User!" because we know him! */
    public int updatertwelcomeback(String name)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues();
        String query,str1;
        int i;
        query="select * from player ";
        Cursor cr = db.rawQuery(query,null);
        if(cr.moveToFirst())
        {
            do{
                str1=cr.getString(0);
                i=cr.getInt(7);
                if(str1.contains(name) && str1.length()==name.length() && i==0)
                {

```

```

        cv.put("username",name);
        cv.put("welcome", 1);
        db.update("player", cv, "username = ?", new String[]{name});
        return 0;
    }
    }while(cr.moveToFirst());
}
return 1;
}
//get Store Photos
public int getStoreflagPhotos(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player "; // select the player table.
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0); // get column
            str4=cr1.getString(1); // get column
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(18); // column 20 where we saved the photo flag
- its numbers of features.
            }
        }while(cr1.moveToNext());
    }
    return -1; // we have no flag.
}
// get string of words for typing game ...
// get password ...
public String getWordsTyping(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4,passing;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0); // user name column
            str4=cr1.getString(1); // password column
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length())
                return cr1.getString(22); // column 24 where we saved the Words
String that we added in Typing Game.
        }while(cr1.moveToNext());
    }
    return ""; // null
}

// updating string of words for typing game ...
public boolean updateWordsTyping(String name,String stringword)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("arlist",stringword);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}

```

```

// Update flag Store Photos
public boolean StoreflagPhotosSave(String name,int storeMath)
{
    SQLiteDatabase db = this.getWritableDatabase(); // writable DB ...
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("flagPhotos", storeMath);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get Store Photos
public int getStorePhotos(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0); // username column
            str4=cr1.getString(1); // password column
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(17); // return according to his location on the
table ...
        }
        }while(cr1.moveToNext());
    }
    return -1;
}

//getStore Math
public int getStoreflagMath(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,str3,str4;
    query="select * from player "; // select from the player ...
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst()) // while the table is not done.
    {
        do{
            str3=cr1.getString(0); // username column
            str4=cr1.getString(1); // password column
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(16); // return according to his location on the
table ...
        }
        }while(cr1.moveToNext());
    }
    return -1;
}
// update flag Store Math - flag = which feature ..
public boolean StoreflagMathSave(String name,int storeMath)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("flagMath", storeMath);
    db.update("player", cv, "name = ?", new String[]{name});
}

```

```

        return true;
    }
    // get Store Math
    public int getStoreMath(String str1, String str2)
    {
        SQLiteDatabase db=getWritableDatabase(); // writable DB ...
        String query,str3,str4;
        query="select * from player "; // select form the table ...
        Cursor cr1 = db.rawQuery(query,null);
        if(cr1.moveToFirst())
        {
            do{
                str3=cr1.getString(0);
                str4=cr1.getString(1);
                if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length())
                    return cr1.getInt(15); // return according to his location on the
table - (17)
            }
            while(cr1.moveToNext());
        }
        return -1; // not found.
    }

    // get flag Store Colors - flag = current feature ...
    public int getStoreflagColors(String str1, String str2)
    {
        SQLiteDatabase db=getWritableDatabase(); // writable DB ...
        String query,str3,str4;
        query="select * from player "; // from player we select
        Cursor cr1 = db.rawQuery(query,null);
        if(cr1.moveToFirst())
        {
            do{
                str3=cr1.getString(0); // user name column
                str4=cr1.getString(1); // password column - we check if we have this
user or no.
                if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length())
                    return cr1.getInt(14); // return according to his location on the
table ...
            }
            while(cr1.moveToNext());
        }
        return -1; // not found.
    }
    // update flag Store Colors - flag = current feature ...
    public boolean StoreflagColorsSave(String name,int storeMath)
    {
        SQLiteDatabase db = this.getWritableDatabase(); // writable DB ...
        ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
        cv.put("name",name);
        cv.put("flagColors", storeMath);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }
    // get Store Colors ...
    public int getStoreColors(String str1, String str2)
    {
        SQLiteDatabase db=getWritableDatabase();
        String query,str3,str4;
        query="select * from player ";

```

```

        Cursor cr1 = db.rawQuery(query,null);
        if(cr1.moveToFirst())
        {
            do{
                str3=cr1.getString(0); // username column
                str4=cr1.getString(1); // password column - we check if we have this
user or no.
                if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                    return cr1.getInt(13); // return according to his location on the
table - (15)
                }
            }while(cr1.moveToNext());
        }
        return -1; // not found
    }
    //getStore Typing
    public int getStoreflagTyping(String str1,String str2)
    {
        SQLiteDatabase db=getWritableDatabase();
        String query,str3,str4;
        query="select * from player ";
        Cursor cr1 = db.rawQuery(query,null);
        if(cr1.moveToFirst())
        {
            do{
                str3=cr1.getString(0); // username column.
                str4=cr1.getString(1); // password column - we check if we have this
user or no.
                if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                    return cr1.getInt(12); // return according to his location on the
table - (14)
                }
            }while(cr1.moveToNext());
        }
        return -1;
    }
    // update flag Store Typing - flag = current feature ...
    public boolean StoreflagTypingSave(String name,int storeMath)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
        cv.put("name",name);
        cv.put("flagTyping", storeMath);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }
    // get Store Typing
    public int getStoreTyping(String str1,String str2)
    {
        SQLiteDatabase db=getWritableDatabase();
        String query,str3,str4;
        query="select * from player ";
        Cursor cr1 = db.rawQuery(query,null);
        if(cr1.moveToFirst())
        {
            do{
                str3=cr1.getString(0); // username column.
                str4=cr1.getString(1); // password column - we check if we have this
user or no.
                if(str1.contains(str3) && str1.length()==str3.length() &&

```

```

str2.contains(str4) && str2.length() == str4.length()) {
    return cr1.getInt(11); // return according to his location on the
table ...
}
}while(cr1.moveToNext());
}
return -1;
}
// get flag Store Clock ... flag = current feature
public int getStoreflagClock(String str1, String str2)
{
    SQLiteDatabase db = getWritableDatabase();
    String query, str3, str4;
    query = "select * from player ";
    Cursor cr1 = db.rawQuery(query, null);
    if (cr1.moveToFirst())
    {
        do{
            str3 = cr1.getString(0); // username column
            str4 = cr1.getString(1); // password column - we check if we have this
user or no.
            if (str1.contains(str3) && str1.length() == str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(10); // return according to his location on the
table - (12)
            }
        }while(cr1.moveToNext());
    }
    return -1; // not found.
}
// update flag Store Clock - flag = feature ...
public boolean StoreflagClockSave(String name, int storeMath)
{
    SQLiteDatabase db = this.getWritableDatabase(); // writable DB
    ContentValues cv = new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name", name);
    cv.put("flagClock", storeMath);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// Updating the Cube two Score ...
public boolean updateScoreCubetwo(String name, int storeMath)
{
    SQLiteDatabase db = this.getWritableDatabase(); // writable DB
    ContentValues cv = new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name", name);
    cv.put("cubetwo", storeMath);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get Store Cube two
public int getScoreCubetwo(String str1, String str2)
{
    SQLiteDatabase db = getWritableDatabase();
    String query, str3, str4;
    query = "select * from player ";
    Cursor cr1 = db.rawQuery(query, null);
    if (cr1.moveToFirst())
    {
        do{
            str3 = cr1.getString(0); // username column

```

```

        str4=crl.getString(1); // password column - checking if the user is
exist.
        if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
            return crl.getInt(21); // return according to his location on the
table - (23)
        }
    }while(crl.moveToNext());
}
return -1; // not found.
}
// Updating the Cube Score ...
public boolean updateScoreCube(String name,int storeMath)
{
    SQLiteDatabase db = this.getWritableDatabase(); // writable DB ...
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("cube", storeMath);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}

// get Store Cube
public int getScoreCube(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,str3,str4;
    query="select * from player "; // select from the player table.
    Cursor crl = db.rawQuery(query,null);
    if(crl.moveToFirst())
    {
        do{
            str3=crl.getString(0);
            str4=crl.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return crl.getInt(20); // return according to his location on the
table ...
            }
        }while(crl.moveToNext());
    }
    return -1;
}

// get Store Clock
public int getStoreClock(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor crl = db.rawQuery(query,null);
    if(crl.moveToFirst())
    {
        do{
            str3=crl.getString(0);
            str4=crl.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return crl.getInt(9); // return according to his location on the
table ...
            }
        }while(crl.moveToNext());
    }
}

```

```

        }
        return -1;
    }
    // update password from Info Layout ...
    public boolean updateCurrentPassword(String name, String pass)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues(); // variable - we use it to insert data
        to the table.
        cv.put("name", name);
        cv.put("pass", pass);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }
    // update score FOCUS Game ...
    public boolean updateScoreFocus(String name, int mScore)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues(); // variable - we use it to insert data
        to the table.
        cv.put("name", name);
        cv.put("focus", mScore);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }

    // get score focus Game ...
    public int getScoreFocus(String str1, String str2)
    {
        SQLiteDatabase db=getWritableDatabase();
        String query,str3,str4;
        query="select * from player ";
        Cursor crl = db.rawQuery(query,null);
        if(crl.moveToFirst())
        {
            do{
                str3=crl.getString(0);
                str4=crl.getString(1);
                if(str1.contains(str3) && str1.length()==str3.length() &&
                str2.contains(str4) && str2.length() == str4.length()) {
                    return crl.getInt(8); // return according to his location on the
                table ...
            }
            while(crl.moveToNext());
        }
        return -1;
    }

    // update score PHOTOS Game ...
    public boolean updateScorePhotos(String name, int mScore)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues(); // variable - we use it to insert data
        to the table.
        cv.put("name", name);
        cv.put("photos", mScore);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }

    // get score photos ...
    public int getScorePhotos(String str1, String str2)
    {

```

```

SQLiteDatabase db=getWritableDatabase();
String query,str3,str4;
query="select * from player ";
Cursor crl = db.rawQuery(query,null);
if(crl.moveToFirst())
{
    do{
        str3=crl.getString(0);
        str4=crl.getString(1);
        if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
            return crl.getInt(7); // return according to his location on the
table ...
        }
    }while(crl.moveToNext());
}
return -1;
}

// Update score COLORS Game ...
public boolean updateScoreColors(String name,int mScore)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("colors", mScore);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get score colors game ...
public int getScoreColors(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor crl = db.rawQuery(query,null);
    if(crl.moveToFirst())
    {
        do{
            str3=crl.getString(0);
            str4=crl.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return crl.getInt(6); // return according to his location on the
table ...
            }
        }while(crl.moveToNext());
    }
    return -1;
}
// update score CLOCK Game ...
public boolean updateScoreClock(String name,int mScore)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("clock", mScore);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get score CLOCK Game ...

```

```

public int getScoreClock(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,str3,str4;
    query="select * from player "; // select from player table.
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(5); // return according to his location on the
table ...
            }
        }while(cr1.moveToNext());
    }
    return -1;
}
// get your password ...
public String getPassword(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length())
                return cr1.getString(1); // return according to his location on
the table ...
        }while(cr1.moveToNext());
    }
    return "";
}
// get email peer user ...
public String getEmail(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length())
                return cr1.getString(19); // return according to his location on
the table ...
        }while(cr1.moveToNext());
    }
    return "";
}
// update score MATH Game ...
public boolean updateScoreMath(String name,int mScore)
{

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv= new ContentValues();// variable - we use it to insert data
to the table.
        cv.put("name",name);
        cv.put("mScore", mScore);
        db.update("player", cv, "name = ?", new String[]{name});
        return true;
    }
// get score math game ...
public int getscoremath(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(2); // return according to his location on the
table ...
            }
        }while(cr1.moveToNext());
    }
    return -1;
}
// update score LETTERS Game ...
public boolean updateletterScore(String name,int letter)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues();// variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("letter",letter);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get score letter ...
public int getScoreLetter(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(3); // return according to his location on the
table ...
            }
        }while(cr1.moveToNext());
    }
    return -1;
}
// update score TYPING Game ...
public boolean updatertypescore(String name,int typescore)

```

```

{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv= new ContentValues(); // variable - we use it to insert data
to the table.
    cv.put("name",name);
    cv.put("typing",typescore);
    db.update("player", cv, "name = ?", new String[]{name});
    return true;
}
// get score Typing Game ...
public int getScoreType(String str1, String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return cr1.getInt(4); // return according to his location on the
table ...
            }
        }while(cr1.moveToNext());
    }
    return -1;
}
// while the user is creating his username - we are checking if the user name is
already exists.
public boolean searchSignUp(String str)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str1;
    query="select * from player ";
    Cursor cr = db.rawQuery(query,null);
    if(cr.moveToFirst())
    {
        do{
            str1=cr.getString(0);
            if(str.contains(str1) && str.length()==str1.length())
                return true; // return true if FOUND.
        }while(cr.moveToNext());
    }
    return false; // NOT FOUND
}
// looking for Email - if we already have it.
public boolean searchEmailSignUp(String mail)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,mail1;
    query="select * from player "; // select from player table.
    Cursor cr = db.rawQuery(query,null);
    if(cr.moveToFirst())
    {
        do{
            mail1=cr.getString(19);
            if(mail.contains(mail1) && mail.length()==mail1.length())
                return true; // return true if FOUND.
        }while(cr.moveToNext());
    }
}

```

```

        return false; // NOT FOUND
    }
    /*while the user is signning his username in - we are checking if the user name is
already exists
and the password is the same password for the user ( if exists ! )*/
public boolean searchSignIn(String str1,String str2)
{
    SQLiteDatabase db=getWritableDatabase();
    String query,str3,str4;
    query="select * from player ";
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            str3=cr1.getString(0);
            str4=cr1.getString(1);
            if(str1.contains(str3) && str1.length()==str3.length() &&
str2.contains(str4) && str2.length() == str4.length()) {
                return true; // return true if FOUND
            }
        }while(cr1.moveToNext());
    }
    return false; // NOT FOUND
}
// here we are looking for the user name and his email.
// if true - then we can log in and the system creates new password which you can
see it at the UserInfo Page.
public boolean searchforgotyourpassword(String user,String mail)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String query,user1,mail1;
    query="select * from player "; // select player table.
    Cursor cr1 = db.rawQuery(query,null);
    if(cr1.moveToFirst())
    {
        do{
            user1=cr1.getString(0);
            mail1=cr1.getString(19);
            if(user.contains(user1) && user.length()==user1.length() &&
mail.contains(mail1) && mail.length() == mail1.length()) {
                return true; // return true if we found the user AND the email ...
            }
        }while(cr1.moveToNext());
    }
    return false; // NOT FOUND
}
// we show here the all info about the user : Name & Scores of all the Levels.
public String get_user_name(String s)
{
    SQLiteDatabase db=getWritableDatabase(); // writable DB ...
    String name=s,name2;
    String all="";
    String query;
    query="select * from player "; // select from player table.
    Cursor cr = db.rawQuery(query,null);
    if(cr.moveToFirst())
    {
        do{
            name2=cr.getString(0);
            if(s.contains(name2) && s.length()==name2.length()) {
                all += name2;
            }
        }return all; // return the name ...
    }
}

```

```
        }
    }while(cr.moveToFirst());
}
return all; // return null if we do not found it ...
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
}
```

Bibliography

- 1.Android Apps for Absolute Beginners**
- 2.hello,android:introducing Google's Mobile Development Platform**
- 3.**
https://www.youtube.com/watch?v=QAbQqLGKd3Y&list=PL6qx4Cwl9DGBsvRxJJOzG4r4k_zLKrnxl
- 4.**
https://www.youtube.com/watch?v=SUOWNXGRc6g&list=PL2F07DBC_DCC01493A
- 5.**
https://www.youtube.com/watch?v=EknElzswvC0&list=PLS1QuiWo1Rlbb1cYyzZpLFCKvdYV_yJ-E