Brain Battle  
מגיש: יהודה אליסף

ת.ז. 328193602

מורה: חגי סוויד

חלופה: תכנות לטלפונים ניידים

תאריך הגשה: 12.6.2023

תוכן

[מבוא 4](#_Toc137076847)

[הרקע לפרויקט 4](#_Toc137076848)

[תהליך המחקר 4](#_Toc137076849)

[טכנולוגיות 4](#_Toc137076850)

[אתגרים מרכזיים 4](#_Toc137076851)

[תיאור תחום הידע 5](#_Toc137076852)

[אובייקטים נחוצים 5](#_Toc137076853)

[סוגי נתונים 5](#_Toc137076854)

[פעולות על המידע 5](#_Toc137076855)

[ארכיטקטורה 6](#_Toc137076856)

[מסכי הפרויקט 6](#_Toc137076857)

[מסך לוגין – מסך הפתיחה 6](#_Toc137076858)

[מחלקות הפרויקט 6](#_Toc137076859)

# מבוא

## הרקע לפרויקט

הפרויקט Brain Battle (קרב מוחות) הוא משחק טריוויה ל2 שחקנים. השחקנים מתחרים ביניהם אונליין, עונים על שאלות טריוויה שונות וצוברים נקודות. בסיום המשחק, השחקנים יכולים לראות סטטיסטיקות שונות כמו מספר התשובות הנכונות והשגויות, ואת מצב הניקוד שלהם ביחס לשחקנים אחרים.

המשחק מיועד לטווח גילאים רחב, משום שיש אפשרות לבחור את רמת הקושי הרצויה (כמובן שרמת הקושי משפיעה גם על הניקוד).להערכתי, תחום הגילאים המומלץ הוא 10 ומעלה.

בחרתי ליצור דווקא משחק כי רציתי ליהנות כמה שיותר מיצירת הפרויקט, ויצירת משחקים מוסיפה גורם של הנאה לפיתוח. בחרתי דווקא בטריוויה כי אני אוהב את המשחק, ונהנה לבחון את הידע הכללי שלי בתחומים שונים.

## תהליך המחקר

כיום, ישנן אפליקציות טריוויה רבות בשוק, גם ל2 שחקנים וגם למספרי שחקנים שונים. עם זאת, זה לא מפריע לי, כי המטרה העיקרית שלי בפיתוח במשחק היא להעמיק את הידע שלי בפיתוח אפליקציות.

## טכנולוגיות

במהלך יצירת הפרויקט השתמשתי במספר טכנולוגיות שאינן חלק מתוכנית הלימודים:

1. HTTP – פרוטוקול תקשורת שמיועד לשליחת וקבלת מידע לאתרי אינטרנט ומהם. השתמשתי בHTTP כדי להוריד שאלות טריוויה ממסד נתונים בשם [Open Trivia Database](https://opentdb.com/), במטרה להשיג כמות גדולה של שאלות בנושאים מגוונים ורמות קושי שונות. פרמטרי כמו מספר השאלות, רמת הקושי וכו' נשלחים בHTTP בעזרת פרמטרי GET.
2. JSON Deserialization – השאלות במסד הנתונים מתקבלות בפורמט JSON. כדי להפוך את השאלות שהתקבלו לאובייקט Question (אובייקט שיצרתי שמייצג שאלה, עליו אפרט בהמשך) יש צורך בדה-סריאליזציה של המידע, כלומר הפיכה שלו מJSON לאובייקט קוד.
3. Firebase – שירות של גוגל שמאפשר מספר פיצ'רים כמו ניהול ההתחברות וההרשמה של משתמשים, ושרת אחסון בשם Firestore שמאפשר לשמור מידע בענן. השתמשתי בFirestore בשביל לשמור את נתונים המשחקים ולסנכרן אותם בזמן אמת, ובשביל לשמור את רשימת המשתמשים ופרטים שלהם.

## אתגרים מרכזיים

היו לי לא מעט אתגרים בזמן העבודה על הפרויקט.

בגלל שהמשחק נערך אונליין בין 2 שחקנים, ואף אחד מהשחקנים לא יכול לשחק בלי השני, הייתי צריך להשתמש ב2 מכשירי פלאפון. לכן, יצרתי אמולטור שמדמה מכשיר פלאפון, והרצתי את האפליקציה גם בו וגם במכשיר הפיזי.

קושי נוסף שהתמודדתי איתו הוא נגרם מכך שהמשחק דורש חיבור אינטרנט רציף. אם באמצע המשחק אחד השחקנים מתנתק, האפליקציה יכולה להיתקע או לקרוס. כדי לפתור את הבעיה, הוספתי לכל תקשורת עם הFirestore פונקציית OnFailure שקובעת מה לעשות במידה וחיבור האינטרנט קרס (למשל: הצגת הודעת שגיאה למשתמש וחזרה למסך הראשי). בנוסף, ההכנסתי את בקשת הHTTP לבלוק try catch, ובמידה וקרתה שגיאה – הודעתי למשתמש שלא ניתן להוריד את השאלות.

# תיאור תחום הידע

## אובייקטים נחוצים

Game: אובייקט הכולל את כל המידע של המשחק. כולל בתוכו את 2 השחקנים ורשימה של שאלות (על מבנה השחקנים והשאלות אפרט בהמשך), ואת הID של המשחק.

User: מייצג משתמש. כולל בתוכו את המידע של המשתמש (שם משתמש, ID וסטטיסטיקות של ניקוד).

Player: מייצג שחקן במשחק. יורש מUser. מוסיף למשתמש רשימה של השאלות הנכונות והשגויות שהוא ענה במשחק.

Question: מייצג שאלה. כולל בתוכו את התשובות השגויות והתשובה הנכונה, קטגוריית השאלה ורמת הקושי שלה.

## סוגי נתונים

בתוך מחלקת Game, השאלות שמורות ברשימה של אובייקטים מסוג Question.

במחלקת Question, התשובות שמורות ברשימה של סטרינגים, והאינדקס של התשובה הנכונה שמור בint. הקטגוריה ורמת הקושי שמורות כל אחת בenum משל עצמה:

enum DifficultyLevel{

EASY,

MEDIUM,

HARD

}

enum Category{

ALL,

GENERAL\_KNOWLEDGE,

SCIENCE,

COMPUTER\_SCIENCE

{

## פעולות על המידע

Game: יצירת משחק, סיום משחק, גטרים וסטרים.

User: יצירת משתמש, מחיקת משתמש, גטרים וסטרים.

Player: יצירת שחקן, חישוב של כמות התשובות הנכונות/שגויות במשחק, מתודת equals(), גטרים וסטרים.

Question: יצירת שאלה, גטרים וסטרים.

# ארכיטקטורה

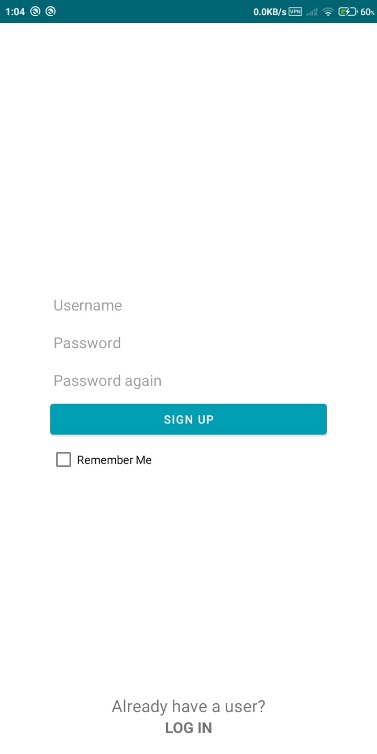
## מסכי הפרויקט

### מסך הרשמה והתחברות (מסך הפתיחה)

במצב התחברות, המסך מציג 2 EditTextים בשביל שם משתמש וסיסמא. במצב הרשמה, המסך מציג EditText נוסף שמיועד להכנסת הסיסמא שוב.

Button

מעבר בין המצבים מתבצע באמצעות לחיצה על הlabel שבתחתית המסך (SIGN UP/LOG IN).

תמונה שמכילה טקסט, צילום מסך, תוכנה, מערכת הפעלה

התיאור נוצר באופן אוטומטי

Button

הרשמה

בודק אם השם משתמש והסיסמא תקינים, ואם כן יוצר משתמש ומעביר למסך יצירת משחק

Button

התחברות

בודק בפיירבייס אם השם משתמש והסיסמא נכונים, ואם כן מעביר למסך יצירת משחק

EditText

הכנסת סיסמא פעם נוספת

EditTexts

הכנסת שם משתמש וסיסמא

CheckBox

קובע האם לשמור את שם המשתמש והסיסמא בהתחברות הבאה

TextView (Clickable)

מעבר ממצב הרשמה למצב התחברות ולהיפך

### מסך יצירת משחק

מאפשר למשתמש ליצור משחק חדש, ולבחור את הקטגוריה, רמת הקושי ומספר השאלות.

אם בפתיחת האפליקציה המשתמש כבר מחובר, זה יהיה מסך הפתיחה.

Buttons

בחירת הקטגוריה של המשחק מבין הבאות:

כל הקטגוריות, ידע כללי, מדע, מדעי המחשב

מסך זה הוא Fragment בתוך הActivity של בתפריט הראשי,

מעבר לFragmentים אחרים מתבצע באמצעות לחיצה על כפתורי הניווט למטה

או החלקה ימינה ושמאלה.

תמונה שמכילה טקסט, צילום מסך, תוכנה, דף אינטרנט

התיאור נוצר באופן אוטומטי

Buttons

בחירת רמת הקושי של המשחק:

קל, בינוני או קשה

TextView

הצגת כמות השאלות:

מינימום 2, מקסימום 20.

Buttons

בחירת כמות השאלות:

כפתור להגדלה וכפתור להקטנה של מספר השאלות

Buttons

בחירת רמת הקושי של המשחק:

קל, בינוני או קשה

Button

מעבר למסך ניקוד

Button

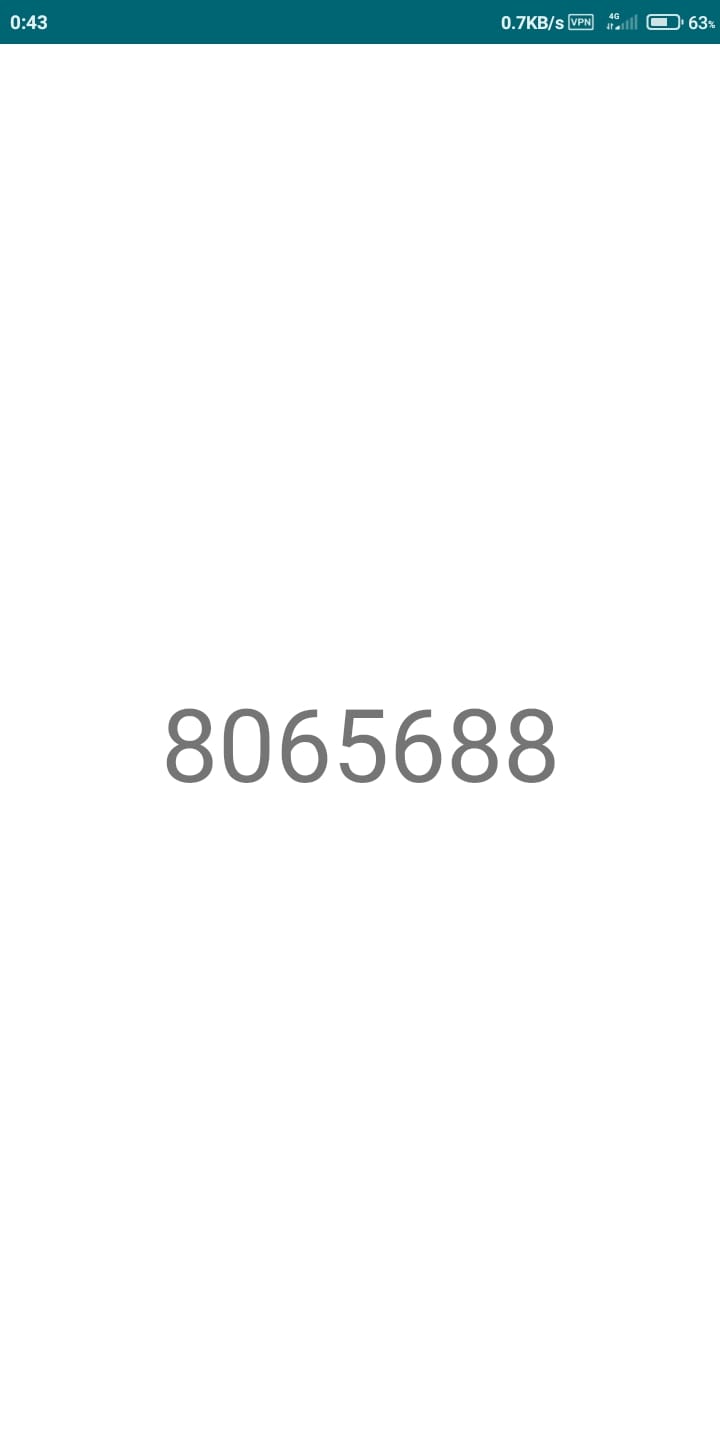
מעבר למסך הצטרפות למשחק

Button

מעבר למסך הגדרות

## מסך קוד המשחק

לאחר יצירת המשחק, יופיע מסך ובו קוד אותו צריך המשתמש השני להכניס כדי להתחבר.

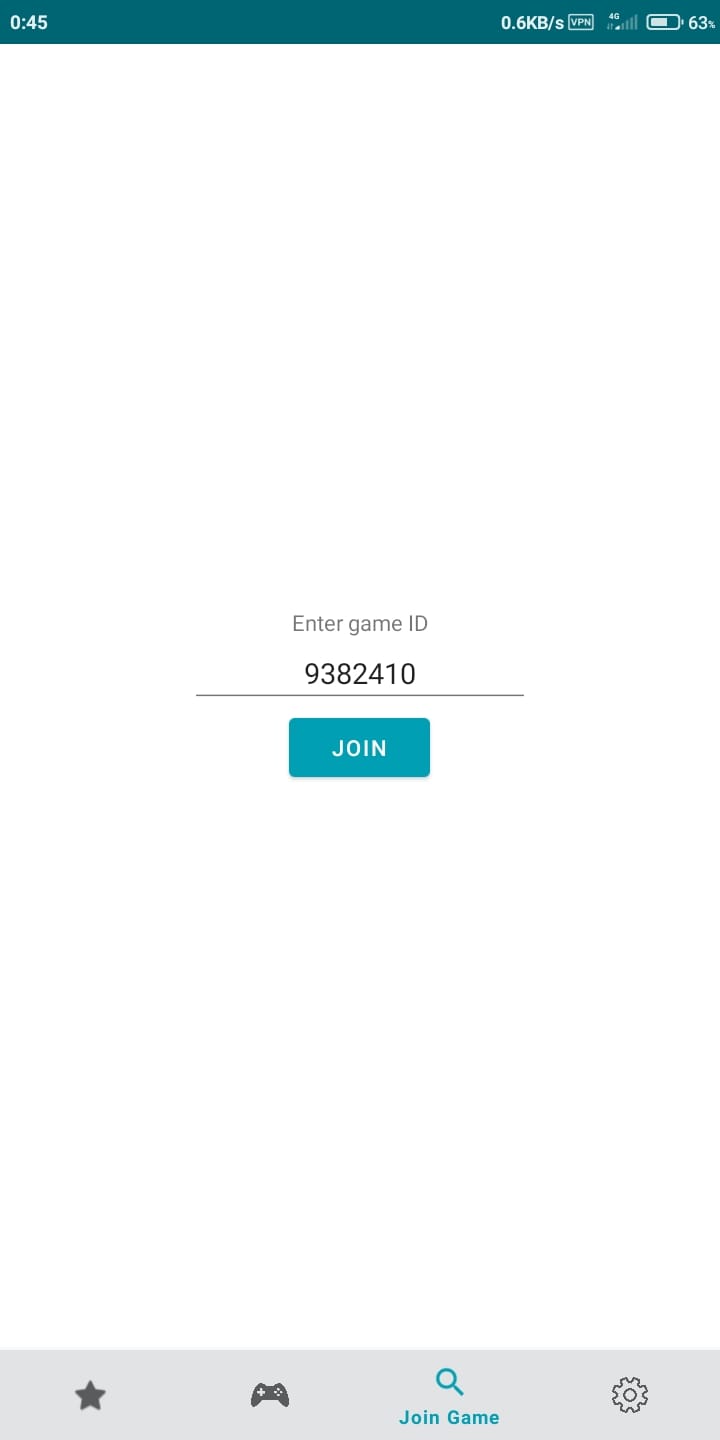


TextView

הצגת קוד המשחק

## מסך הצטרפות למשחק

אחרי שהשחקן הראשון יצר את המשחק, השחקן השני יכניס במסך זה את קוד המשחק כדי להצטרך אליו.



Button

הצטרפות למשחק

EditText

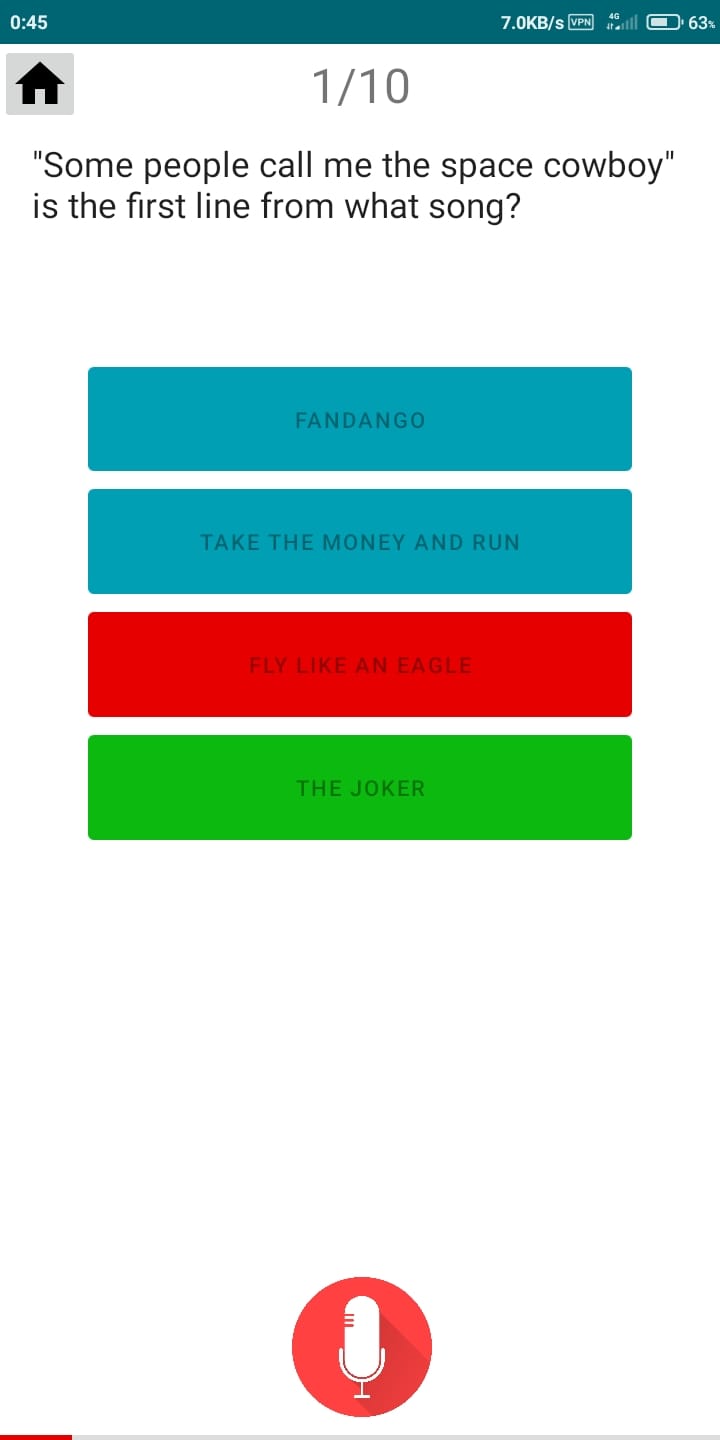
הכנסת קוד המשחק

## מסך המשחק

כששני השחקנים מחוברים, הם מועברים למסך זה, בו המשחק עצמו מתנהל.

Button

חזרה לתפריט הראשי



ImageButton

הקלטת תשובה

המשתמש צריך ללחוץ על הכפתור ולומר את מספר התשובה

(one, two, three, four)

הפיצ'ר עדיין לא ממומש

Canvas

הצגת ההתקדמות של היריב והאם הוא צדק או טעה (ירוק/אדום בהתאמה)

TextView

הצגת השאלה

TextView

הצגת מספר השאלה הנוכחית וסך השאלות

Buttons

כפתורי תשובות

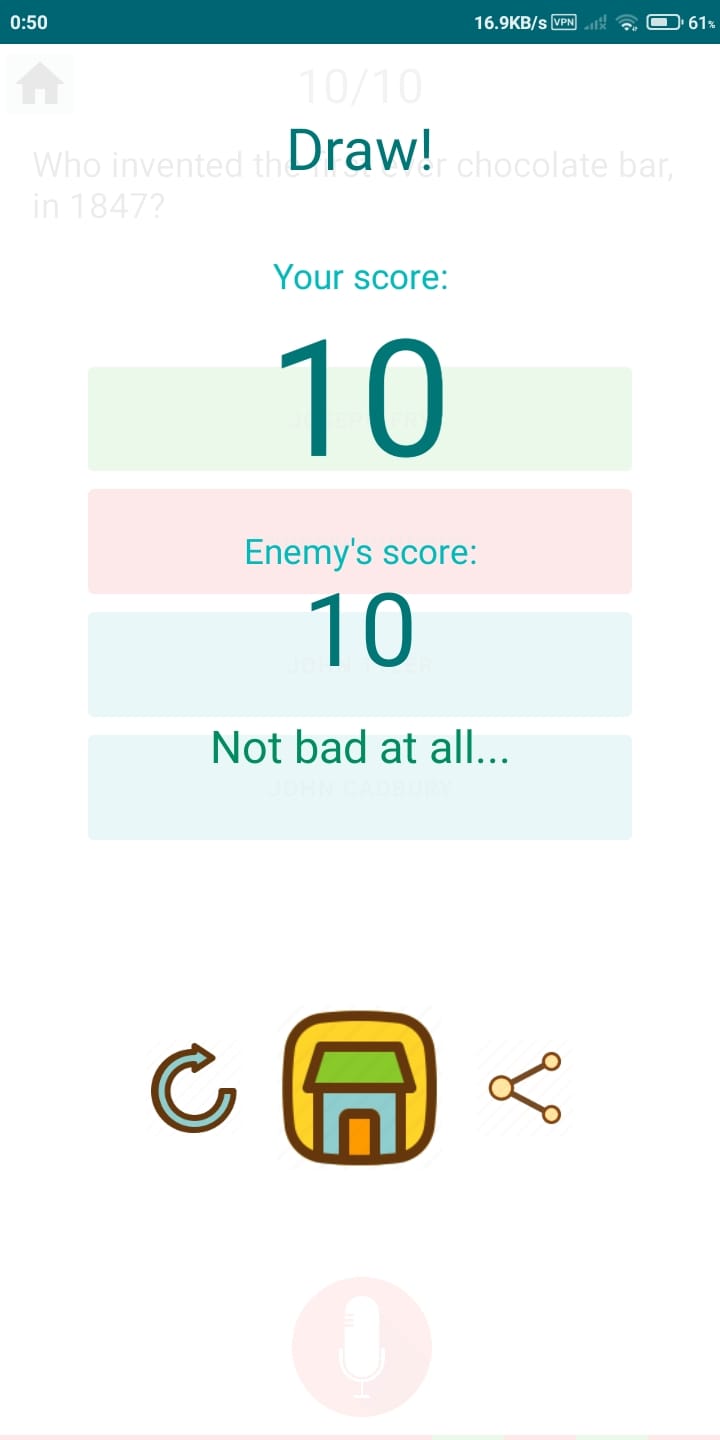
המשתמשים צריכים ללחוץ על התשובה הנכונה. לאחר מכן, הכפתור ייצבע באדום או ירוק בהתאם לתשובה

## מסך סיום משחק

בסוף המשחק, יופיע מסך זה, המציג את הניקוד ואת המנצח במשחק.

TextView

הצגת השחקן המנצח



ImageButton

שיתוף תוצאות משחק

ImageButton

חזרה למסך הבית

ImageButton

יצירת משחק חדש עם אותם פרמטרים (כמות שאלות וכו')

TextView

הצגת הניקוד של היריב

TextView

הצגת הניקוד שלך

## מסך ניקוד

במסך זה המשתמש יכול לראות את מצב הנקודות שלו ושל אחרים, וסטטיסטיקות נוספות.

TextView

סה"כ תשובות נכונות

TextView

סך הניקוד שלך

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי

ListView

הצגת הניקוד שלך מול משתמשים אחרים

ProgressBar

הצגת אחוז התשובות הנכונות מבין כל התשובות

TextView

סה"כ תשובות שגויות

## מסך הגדרות

במסך זה אפשר לערוך את הגדרות האפליקציה והגדרות המשתמש.

TextView

שם המשתמש

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

ImageButton

שינוי שם המשתמש

ImageButton

מחיקת המשתמש

ProgressBar + ImageButton

שינוי עוצמת מוזיקת הרקע

שם המשתמש

## תרשים מסכים

LoginActivity

## 

MainMenuActivity

GameIdFragment

GameActivity

EndGameFragment

NewGameFragment

JoinGameFragment

SettingsFragment

ScoreFragment

## מחלקות הפרויקט

