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.

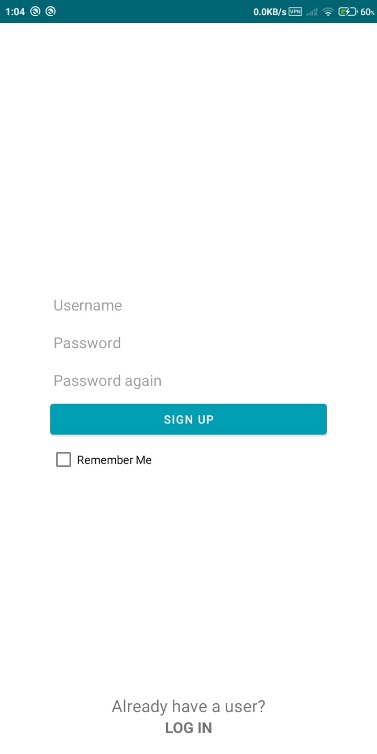
# ארכיטקטורה

## 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.

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

### מסך יצירת משחק

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

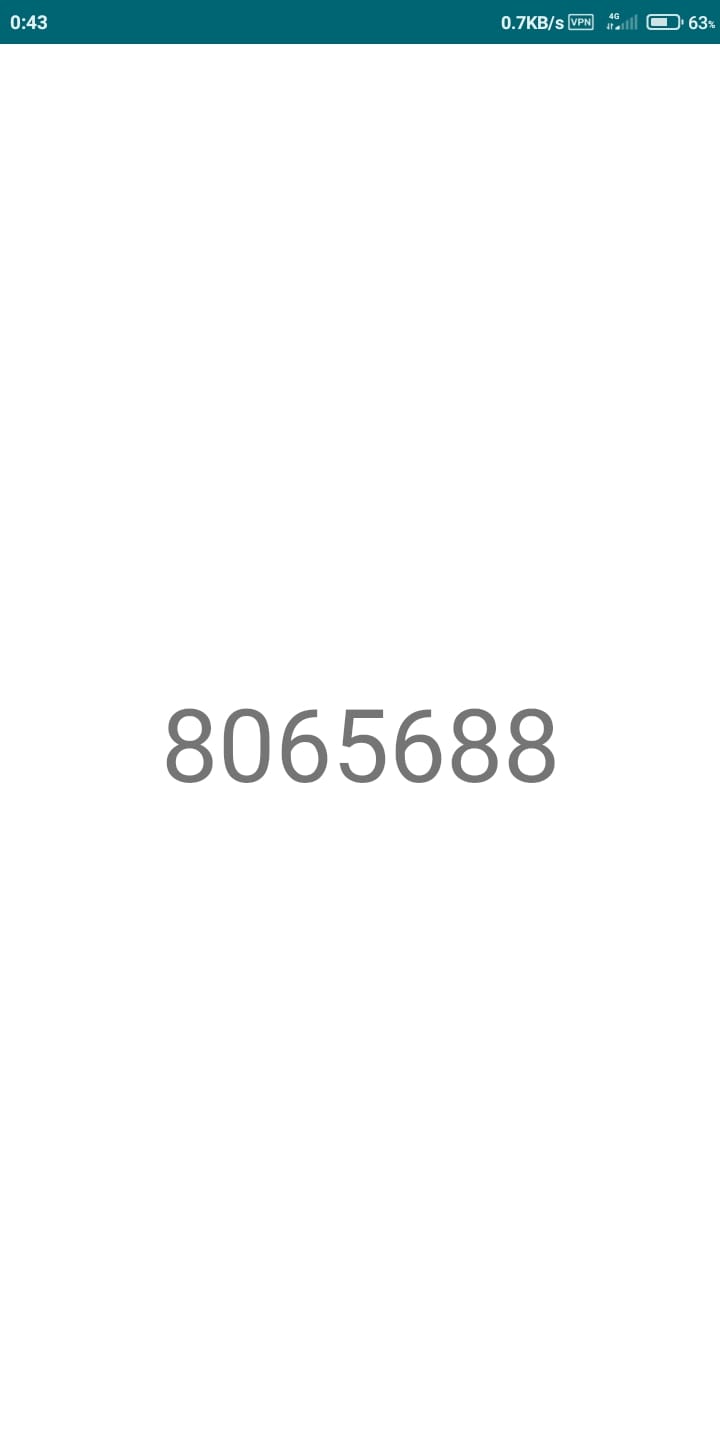
אם בפתיחת האפליקציה המשתמש כבר מחובר, זה יהיה מסך הפתיחה.

תמונה שמכילה טקסט, צילום מסך, תוכנה, דף אינטרנט

התיאור נוצר באופן אוטומטי

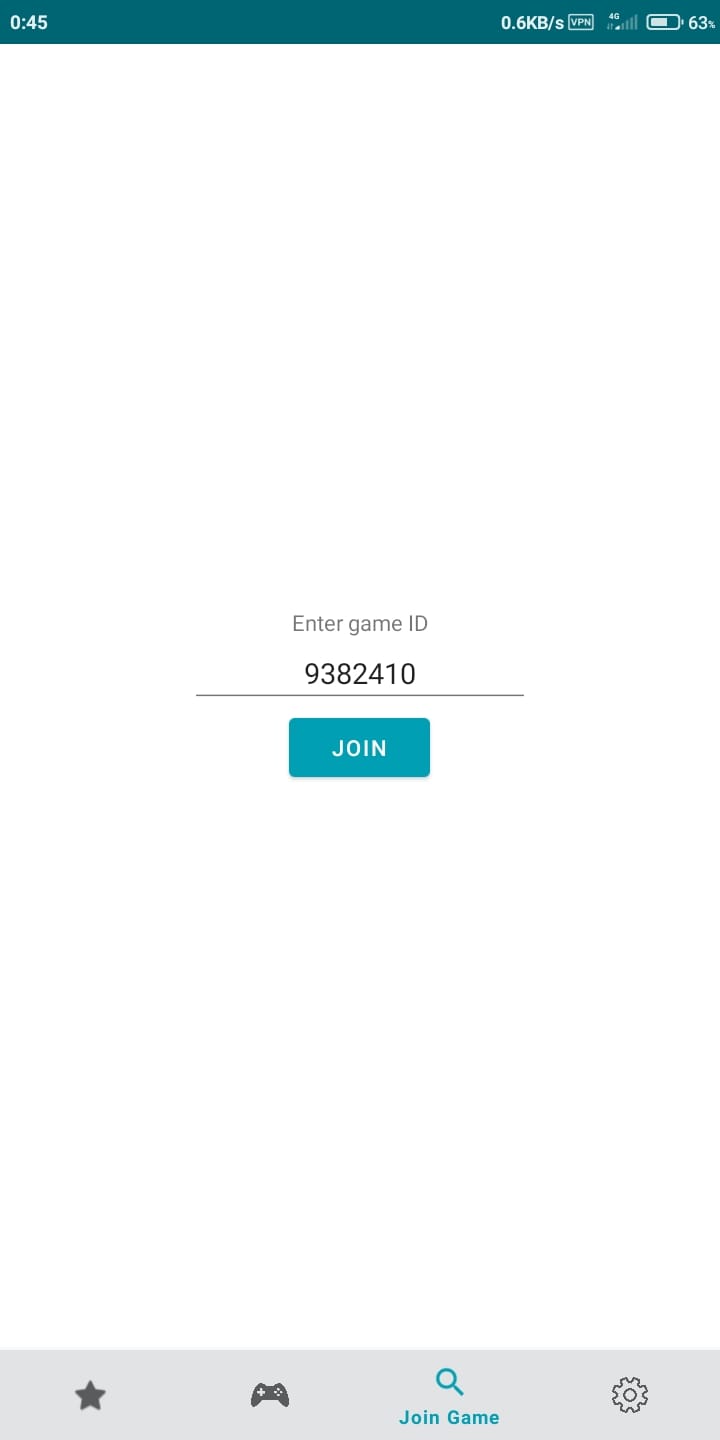
## מסך קוד המשחק

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



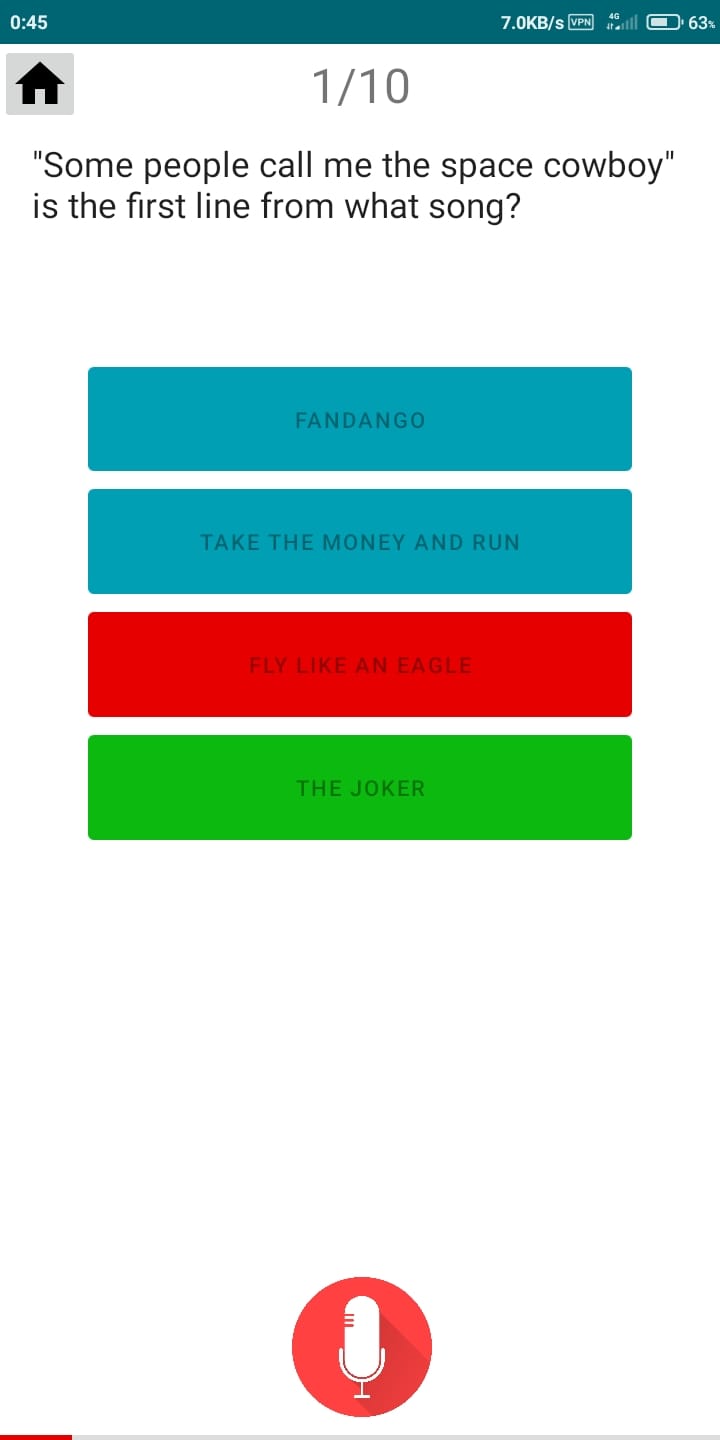
## מסך הצטרפות למשחק

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



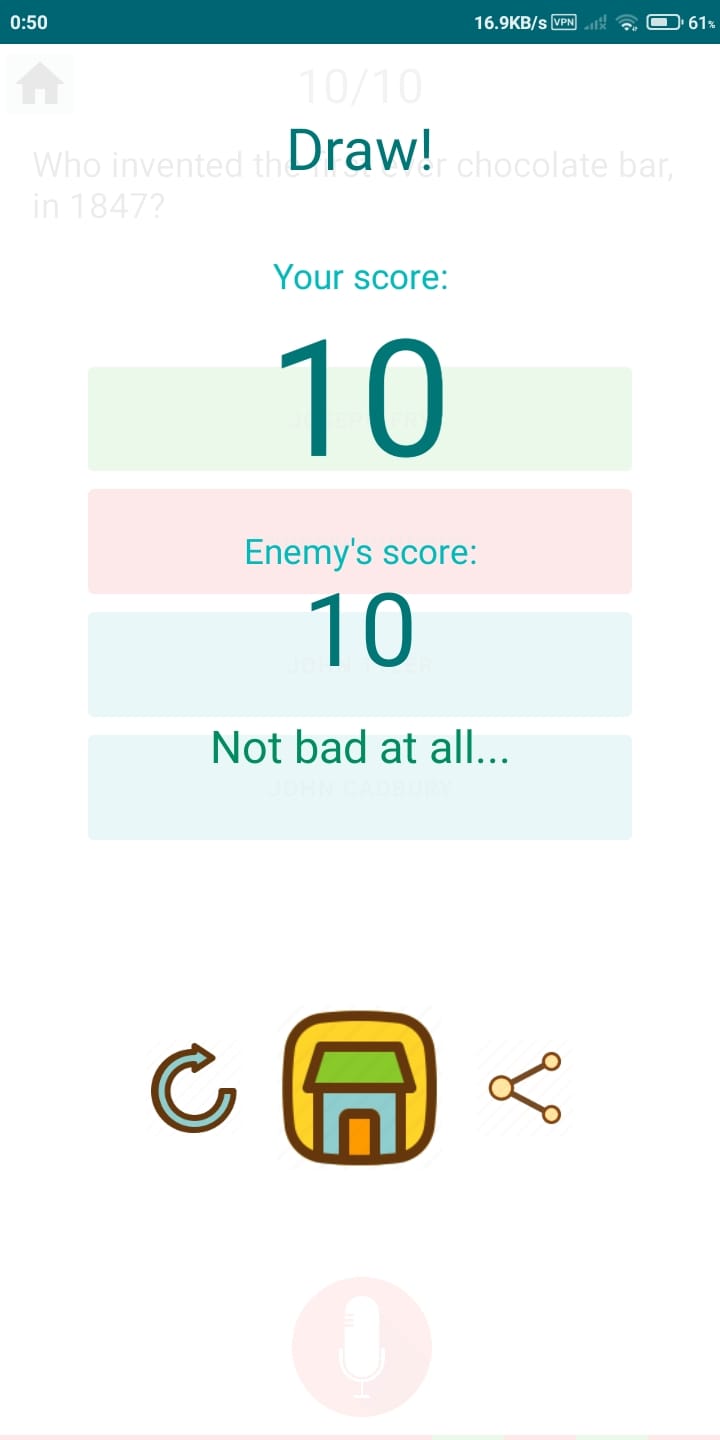
## מסך המשחק

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



## מסך סיום משחק

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



## מסך ניקוד

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

תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטי

## מסך הגדרות

במסך זה אפשר לערוך את הגדרות האפליקציה והגדרות המשתמש.

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

## מחלקות הפרויקט