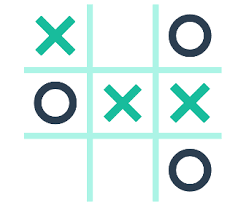
**ספר פרויקט גמר י"ב**

**שם האפליקציה: Tik Tak Tok**

****

**שם המגיש/ה: שירה לוינסון**

**תעודת זהות: 326173309**

**שנת הלימודים: 2022**

**בית ספר: קציר – על יסודי משגב (640565)**

**שם המנחה: הדס יגאל BSc הנדסת תוכנה**

**תוכן עניינים**

**תוכן עניינים**

[**מבוא 3**](#_Toc105595474)

[**מדריך למשתמש 4**](#_Toc105595475)

[**ברקוד להורדת האפליקציה: 4**](#_Toc105595476)

[**תיאור מסכי האפליקציה 5**](#_Toc105595477)

[**תרשים מחלקות UML 9**](#_Toc105595478)

[**תרשים זרימה 10**](#_Toc105595479)

[**עמידה בדרישות הפרויקט check list 11**](#_Toc105595480)

[**הסבר מפורט על קטעי קוד 13**](#_Toc105595481)

[**מבנה הנתונים בפיירבייס 23**](#_Toc105595482)

[**קבצים נלווים-תמונות, צבעים ומחרוזות 28**](#_Toc105595483)

[**סיכום 29**](#_Toc105595484)

[**נספחים- רשימות תוכנה מלאות 30**](#_Toc105595485)

# 

# מבוא

**האפליקציה שהכנתי היא משחק איקס עיגול. בחרתי לממש את המשחק הזה מכיוון שתמיד אהבתי לשחק אותו עם חברים, וחשבתי שיהיה מהנה מאוד לממש את האלגוריתם שעומד מאחוריו. בנוסף המשחק שלי מאפשר לחברים לשחק אותו אחד עם שני מרחוק.  
איקס עיגול (או בשמותיו באנגלית: noughts and crosses = בריטית, או tic-tac-toe = אמריקאית), הנקרא גם איקס מיקס דריקס הוא משחק לשני שחקנים המשוחק על לוח של 3X3 משבצות. כל שחקן בתורו מסמן סימן באחת המשבצות (לרוב איקס או עיגול, ומכאן שם המשחק) השחקן המנצח הוא זה שהצליח למלא שורה, טור או אלכסון בסימנים שלו. אם אף שחקן לא הצליח לעשות זאת עד שכל המשבצות מלאות, המשחק נגמר בתיקו.**

**בדרך כלל נהוג להתחיל במשבצת קיצונית וזאת כדי ליצור בהמשך שורה וטור (או שורה ואלכסון, וכדומה) שבכל אחד מהם שני סימונים, כאשר אם יבחר השחקן השני למנוע אחד מהם, ינצח השחקן המתחיל בעזרת שני הסימונים האחרים (לדוגמה: השחקן ראשון שם ב-A1, A3, ו-C3). בשל טכניקות אלו בדרך כלל מגיע השחקן שהתחיל שני למצב של הפסד או תיקו.**

**במונחי תורת המשחקים, המשחק הוא משחק פתור: לשני השחקנים במשחק יש תכסיס תיקו. כלומר, כל שחקן יכול במקרה הגרוע להבטיח לעצמו תיקו אם ישחק משחק מושלם. לפיכך, אם שני השחקנים משחקים בצורה הטובה ביותר, הם יגיעו לתיקו.**

**באפליקציה שלי כל שחקן יכול לשחק עם חבר, בשני טלפונים שונים, והאפליקציה תשמור את תוצאותיהן של השחקנים.**

**מאז שחשבתי סתם על הרעיון לעשות את המשחק הזה באפליקציה עברו שנתיים ואני בכיתה יב' ונודע לנו שצריך לעשות פרויקט שהוא לבנות אפליקציה משלך ואז ישר הכל התחבר לי והתחלתי לעבוד. באפליקציה שלי ממש אפשר לראות את המשחק מבוסס אבל עם שינויים משלי, כל משתמש שנכנס לאפליקציה מתחיל במסך הראשי שבו הוא יכול כל פעם מחדש לשים לעצמו איזה שם שבא לו.**

**ואז יש לו שתי אופציות ליצור חדר משחק או להצטרף לאחד קיים דרך קוד. אם ובחר ליצור חדר משחק אז הוא יוכל לשחק עם ההגדרות של אותו משחק, לבחור עם איזה שחקן לשחק ומי לאתחל את הלוח. ואם בחר להצטרף לחדר משחק קיים דרך קוד הוא פשוט יחכה שמי שיצר את אותו החדר יתחיל את המשחק.**

**בשביל להתחיל משחק חייב 2 משתתפים בלבד. משתמש גם לא יוכל להכנס לחדר משחק אם השם שבחר כבר קיים או אם החדר מלא.**

**פעולות נוספות של האפליקציה:**

* **מוזיקת רקע**
* **התראה על חוסר חיבור לאינטרנט**

**בספר זה אציג את המסמכים, המחלקות, מבנה נתונים של האפליקציה וגם אסביר כיצד להשתמש בה.**

**סביבת עבודה: android studio**

**גרסת האנדרואיד סטודיו: 4.1**

**שפת תכנות: java**

**קישור לרשימות תוכנה מלאות באתר גיטהב:** [**https://github.com/ShiraLev100/Yud-Bet-Project.git**](https://github.com/ShiraLev100/Yud-Bet-Project.git)

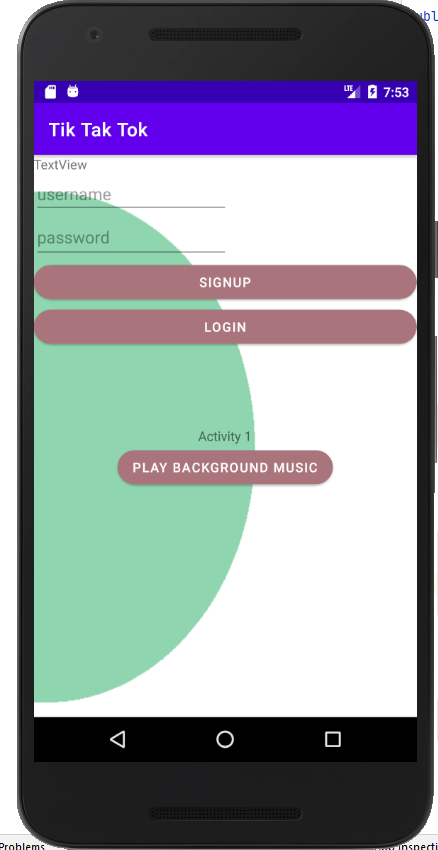
# מדריך למשתמש

# ברקוד להורדת האפליקציה:

****

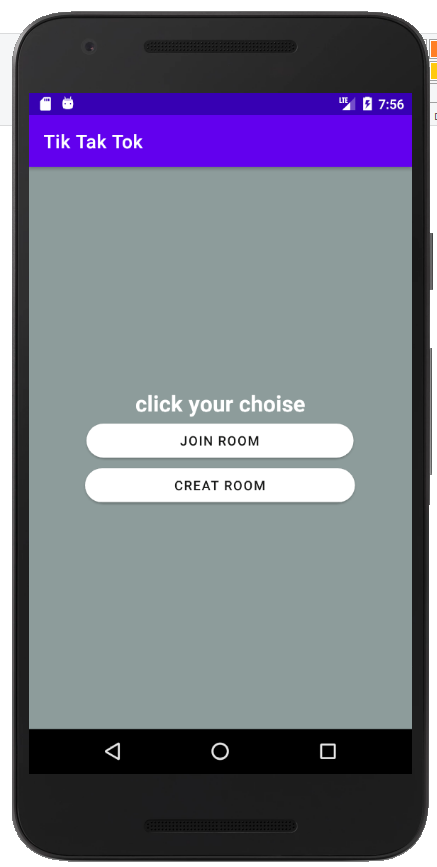
# תיאור מסכי האפליקציה

**מסך הבית-** בו אפשר להתחבר למשתמש קיים או ליצור משתמש חדש ולבחור בתפריט האם להפעיל מוזיקת רקע שתפעל בזמן המשחק.

****

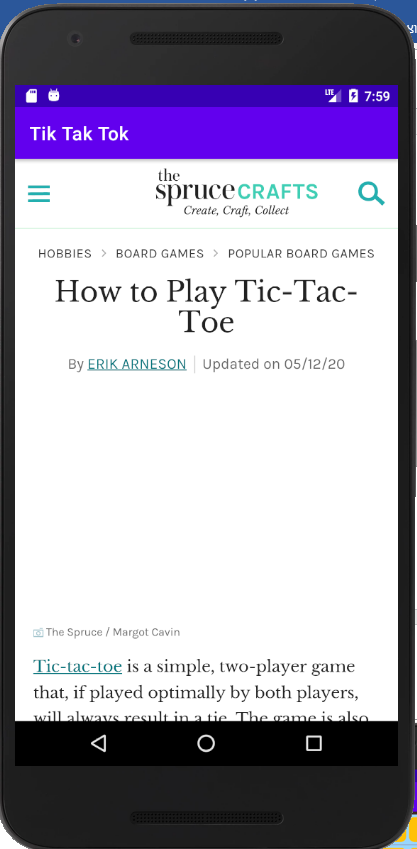
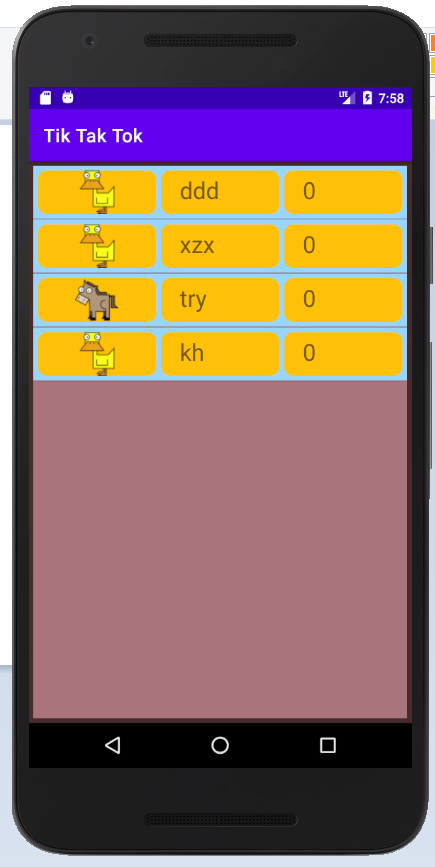
**מסך חדר משחק-** בו אפשר לראות את הגדרות המשחק, את המשתתף שהצטרף ולשחק איקס עיגול. ניתן לאתחל את המשחק בכפתור reset או לעבור לשלושה מסכים אחרים דרך שלושת הנקודות למעלה

**מסך בחירת חדר משחק-** בו אפשר לבחור האם להצטרף לחדר משחק קיים או ליצור חדר משחק חדש ולחכות לחבר.

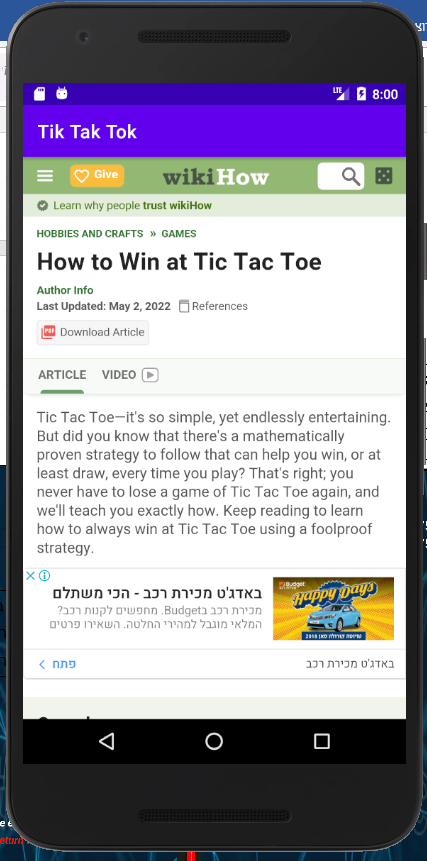
****

**מסך טבלת הדירוג הסופי-** בו אפשר לראות את הניקוד הכללי והסופי של כל אחד מהמשתתפים.

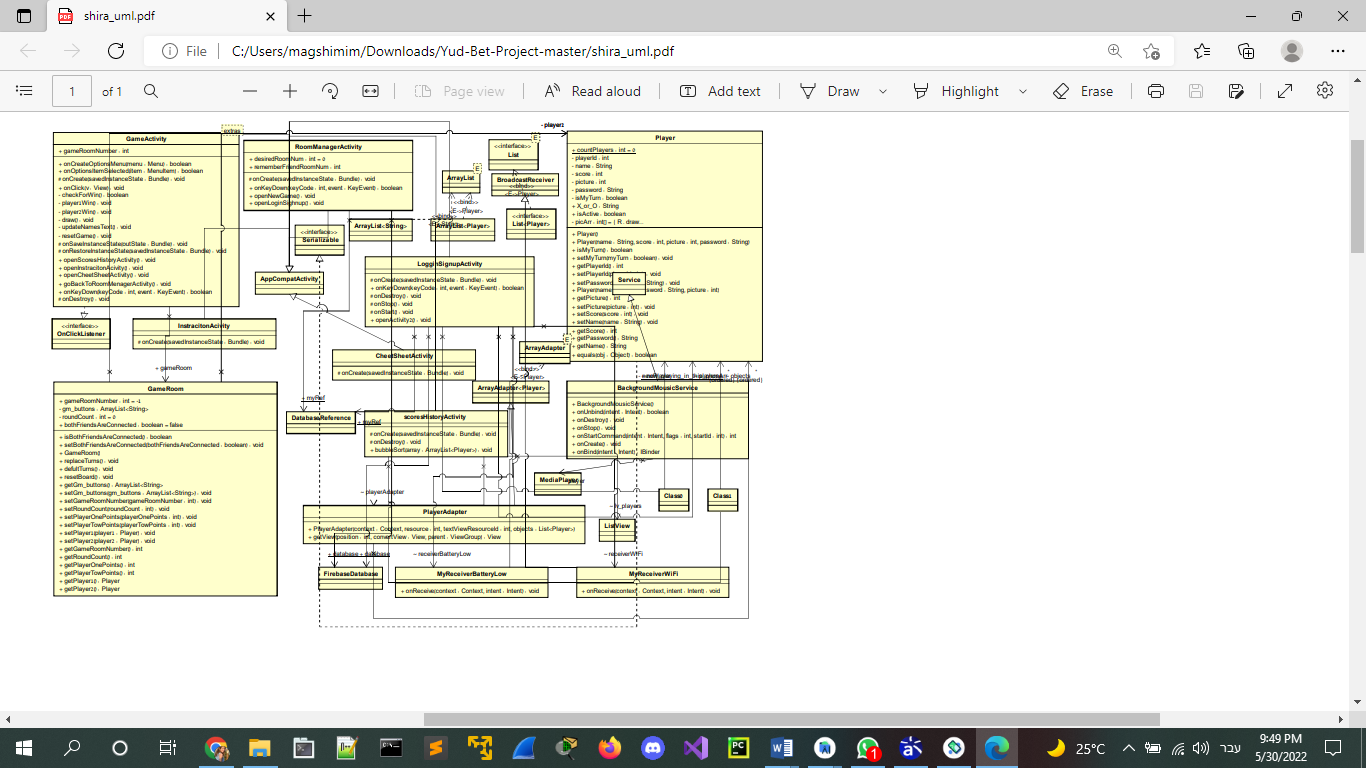
**מסך חוקי משחק איקס עיגול-** המסך מציג אתר שמגולל את חוקי המשחק איקס עיגול

****

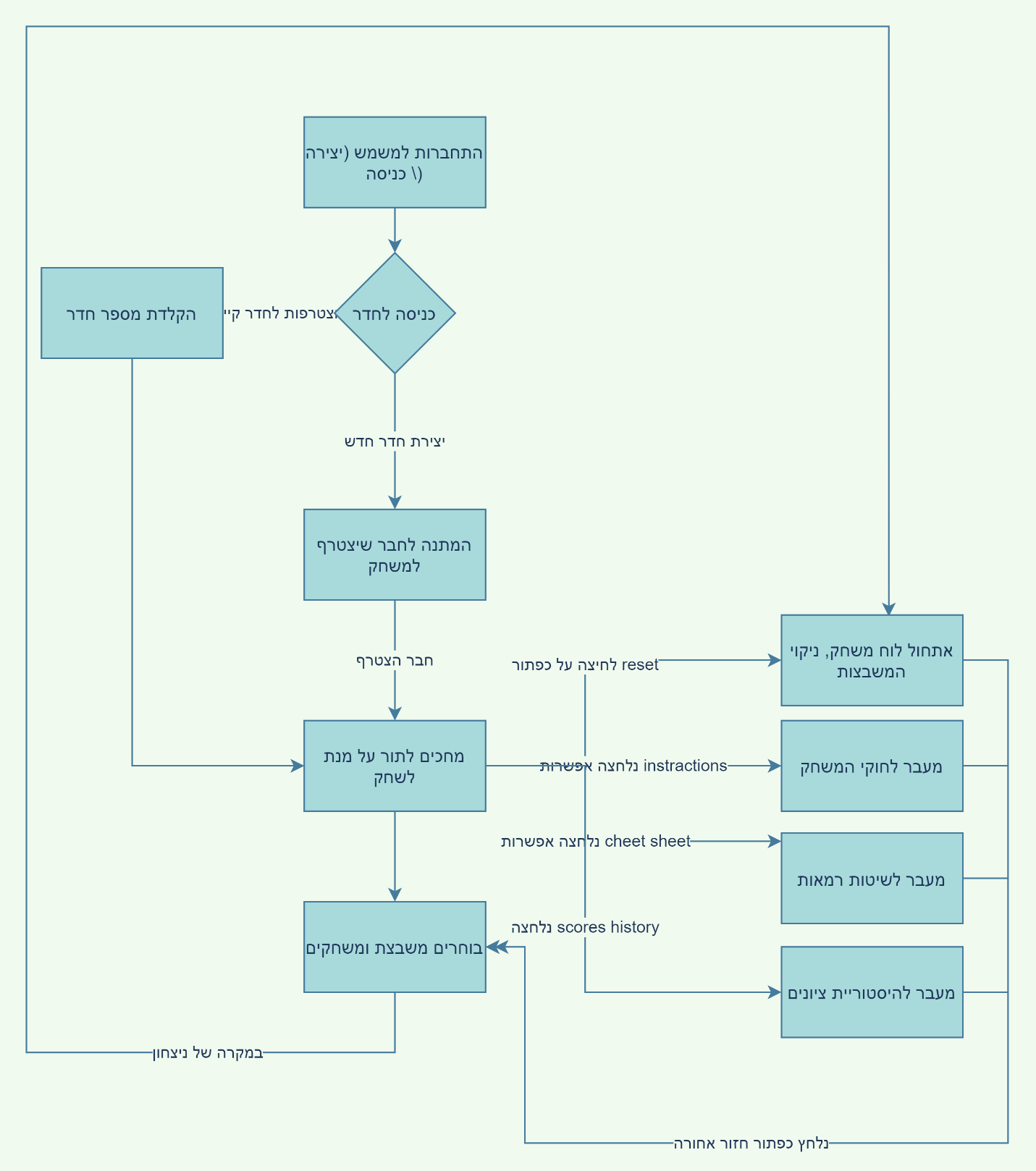
Cheet sheet- מסך אשר מגולל דרכים שונות לנצח במשחק, מראה את הדרכים בהן אפשר לרמות.

****

# תרשים מחלקות UML

****

# תרשים זרימה

****

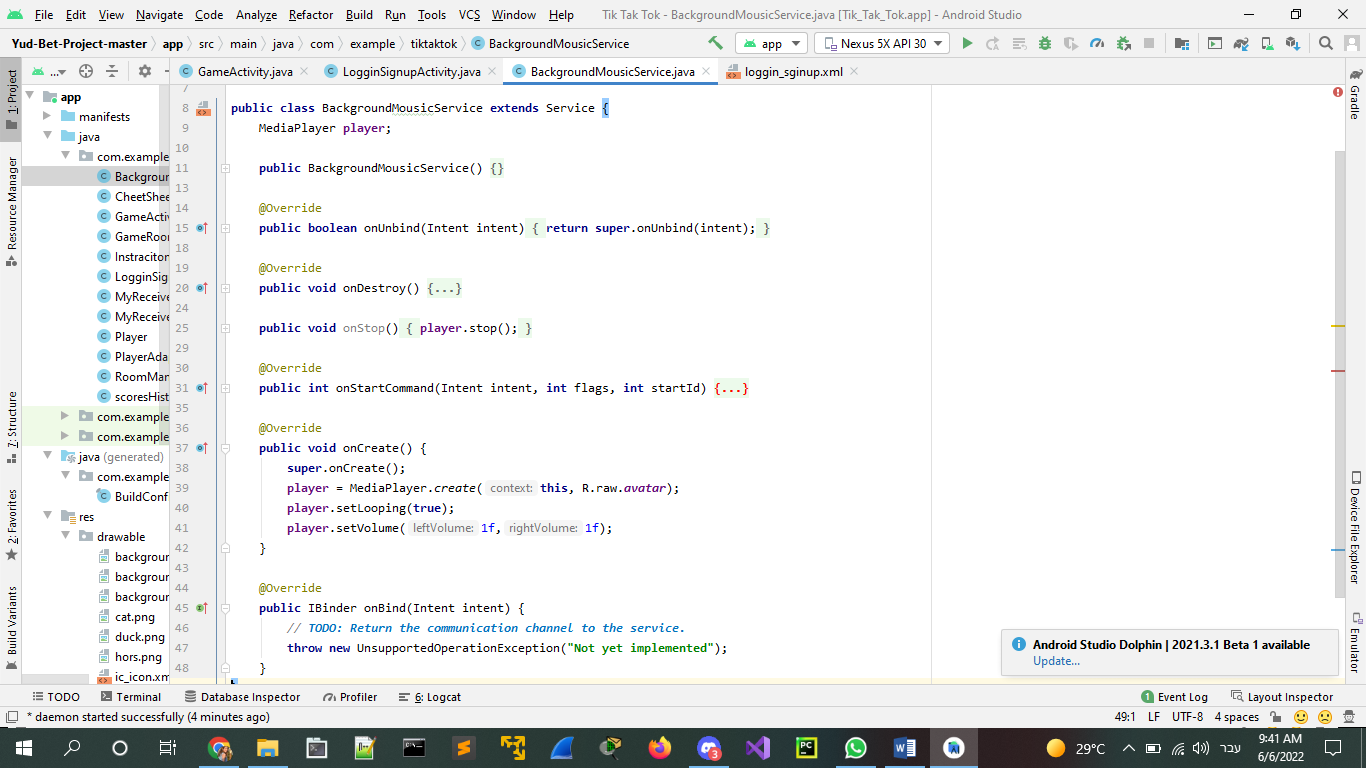
# עמידה בדרישות הפרויקט check list

|  |  |
| --- | --- |
| **הדרישה** | **פירוט מענה על הדרישה בפרויקט** |
| **לפחות 5 מסכים שונים** | **המסכים שלי הם:** |
| **לפחות 5 מחלקות** | **המחלקות שלי הן:** |
| **שימוש בתפריטים** | **תפריטים אצלי בפרויקט:**  **מימשתי תפריט שמאפשר לעבור לשלושה מסכים- מך הוראות משחק, מסך רמאות ומסך היסטוריית שחקנים.** |
| **שימוש בתיבות דו שיח** | **תיבות דו שיח אצלי בפרויקט:**  **הדיאלוג משמש להזנת קוד החדר אליו מעוניינים להצטרף.** |
| **Service** | **אצלי בפרויקט מימשתי את הservice הבא:**  **backgroundMusicService** |
| **Intent** | **אצלי בפרויקט הintent הבאים:**  **בשימוש למעבר בין חלונות והפעלת הסרביס.** |
| **רכיבים מתקדמים חובה לפחות אחד מהרשימה הבאה:**  **· גרפיקה**  **· אנימציה**  **· שימוש במצלמה של הטלפון**  **· טלפוניה: חיוג, קבלת שיחה נכנסת, קבלת / שליחת SMS**  **· חיישני תנועה**  **· חיישני מיקום ומפות**  **· מיקרופון** | **אצלי בפרויקט:**  **-גרפיקה**  **יש לי מסך צייר כולל בחירת צבע ועובי קו ובנוסף אני מציג את תמונות של כל הציורים של כל המשתמשים.**  **-אנימציה**  **בעת ההתחברות של המשתמש אני מציגה את האווטאר הנבחר שלו, והוא נע מצד לצד כאשר המשתמש לוחץ על כפתור התחברות.** |
| **לפחות אחד משני הנושאים הבאים:**  **Broadcast Receiver (= מחלקה אשר יכולה להאזין למסרים) לדוגמא:**  **· האזנה למצב הבטריה**  **· האזנה לשיחה נכנסת או יוצאת**  **· האזנה לשעון**  **· האזנה לכניסה של sms**  **Content provider (= ספקי תוכן שמאפשרים שיתוף נתונים בין אפליקציות. במידה ואתם מעוניינים שהאפליקציה שלכם תשתף נתונים עם אפליקציות אחרות אזי נפתח ספק תוכן)**  **לדוגמא :**  **· אנשי קשר**  **· יומן שיחות**  **· הגדרות התקן**  **· יומן לוח שנה**  **· ועוד** | **דוגמא לBroadcast Receiver בפרויקט שלי:**  **האזנה למצב האינטרנט - MyReceiverWiFi**  **האזנה למצב הבטריה- MyReceiverBatteryLow** |

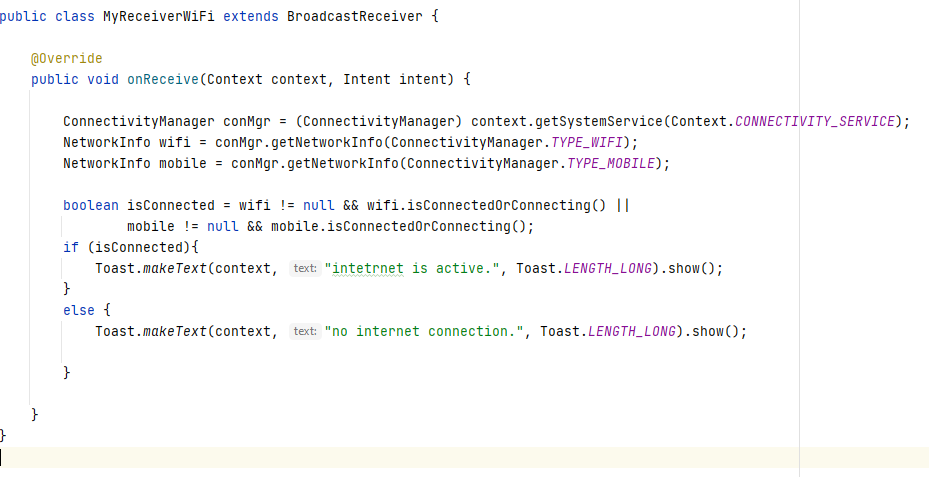
# הסבר מפורט על קטעי קוד

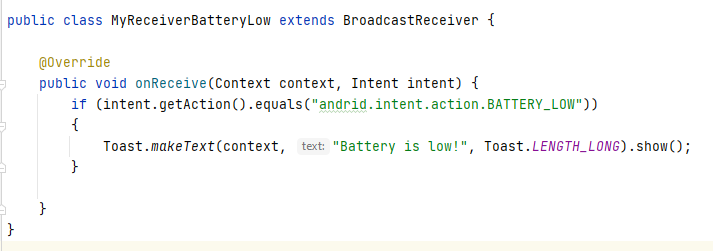
1. **Service**

**בפרויקט השתמשתי בservice בשביל מוזיקת רקע לאפליקציה גם כאשר המסכים מתחלפים או משתנים המוזיקה ממשיכה להתנגן.**

****

1. **Broadcast Receiver**

**זו יכולת של האנדרואיד להאזין כל הזמן ובכל המסכים של האפליקציה לאירועי מערכת שמעניינים אותנו. בפרויקט שלי ה Broadcast Receiver משמש לבדיקת מצב האינטרנט של הטלפון או את אחוז הבטרייה, במידה ואין חיבור לאינטרנט או סוללה נמוכה ובמידה ויש תופיעה הודעה למשתמש.**

****

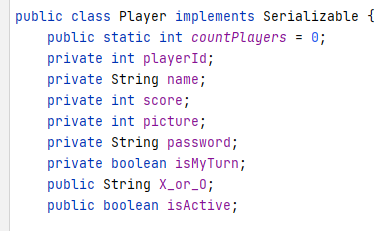
1. **מבנה נתונים עיקריים**

**המחלקות המהותיות ביותר הןPlayer וה-GameRoom**

**Player:**

**וכמובן יש למחלקה פונקציה בונה setter ו- getter.**

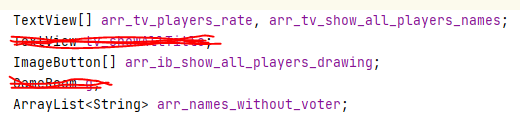
**ניתן לראות בקוד שכל שחקן יודע מה הניקוד שלו, מה השם שלו ומה התמונה שלו (שנבחרת באופן רנדומלי) מה הסיסמה שלו ומתי זה תורו לשחק, והאם הוא פעיל (מחובר) או לא.**

****

**: GameRoom**

**ניתן לראות בקוד את התכונות והפעולות של חדר המשחק.**

****

****

**4) הסרת חדר משחק שלם מהפיירבייס**

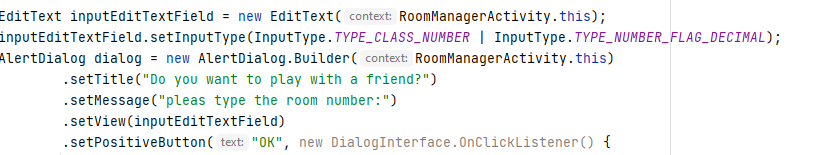
ניתן לראות בקוד שכאשר המשתמש יוצא מהמשחק פועלת הפעולה להסרת חדר המשחק מהפיירבייס.

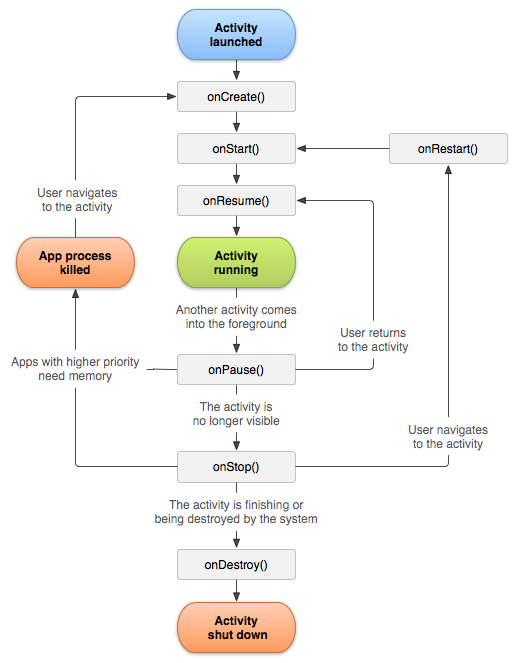
מה שהפעולה עושה זה בעצם הולכת לכל המשחקים הקיימים בפיירבייס מחפשת את מספר החדר הנוכחי ומסירה את הערך שלו מהפיירבייס.

****

**5) כתיבת טקסט בתוך דיאלוג**

**ניתן לראות בקוד איך נוצר תיבת טקסט בתוך הדיאלוג.**

****

**6) activity life cycle**

האיור בתמונה מציג את מחזור החיים של ה activity-, בפרויקט שלי התעוררה בעיה של ריענון תצוגה לא רצוי משהו שגרם להבהובים, כאשר הייתי מנסה לחזור אחורה למסך קודם. כשדיבגתי את הבעיה הבנתי שהבעיה נוצרת משום שאני ממשיכה להתחבר לאותו משתמש גם כאשר אני מנסה להתחבר למשתמש אחר. פתרתי את הבעיה בעזרת עדכון הפעילות של המשתמש כאשר אני חוזרת אחורה, כלומר בonDestroy.

**7) Custom array adapter**

**משמעות המילה adapter היא מתאם. מטרת המתאם היא להציג בlistView- את המידע שנמצא במערך, כלומר לתאם בין מערך שהוא מבנה נתונים רעיוניים לבין ה-listView שנמצא על המסך שהמשתמש יכול לצפות בו.**

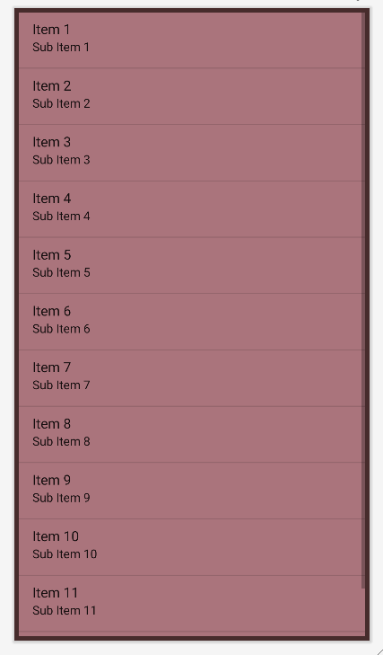
**בפרויקט שלי אני מציג את השמות, הנקודות, התמונה והמקום (מקום כלומר מקום ראשון, מקום שני וכו') של השחקנים.**

****

**שלבי העבודה שלי היו:**

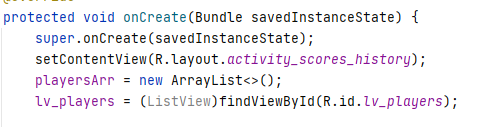
**שלב 1:**

**לבנות קובץ xml שמתאר מה המבנה של כל שורה באותה טבלה.**

****

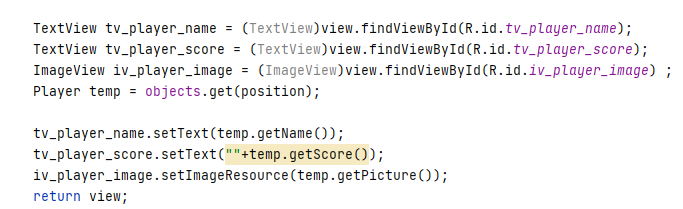
**שלב 2:**

**הכנתי מערך שבו אני רוצה להציג את הנתונים בטבלה.**

****

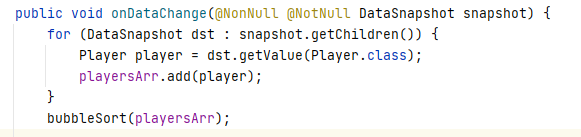
**שלב 3:**

**כתבתי קובץ java שמסביר איך הadapter ממלא את השורות של הטבלה.**

****

**שלב 4:**

**אני ממיינת את השורות שיש בטבלה לפי הניקוד שיש לשחקנים.**

****

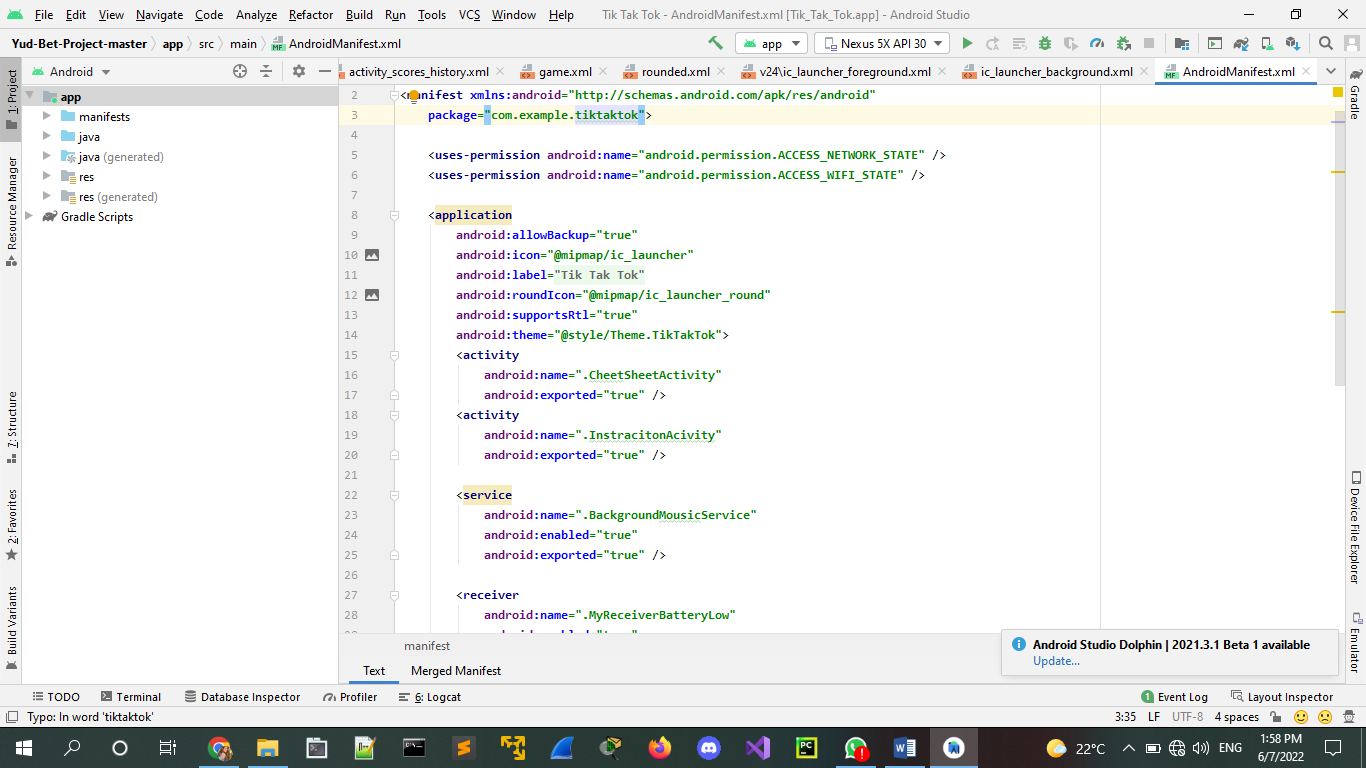
**שלב 5:**

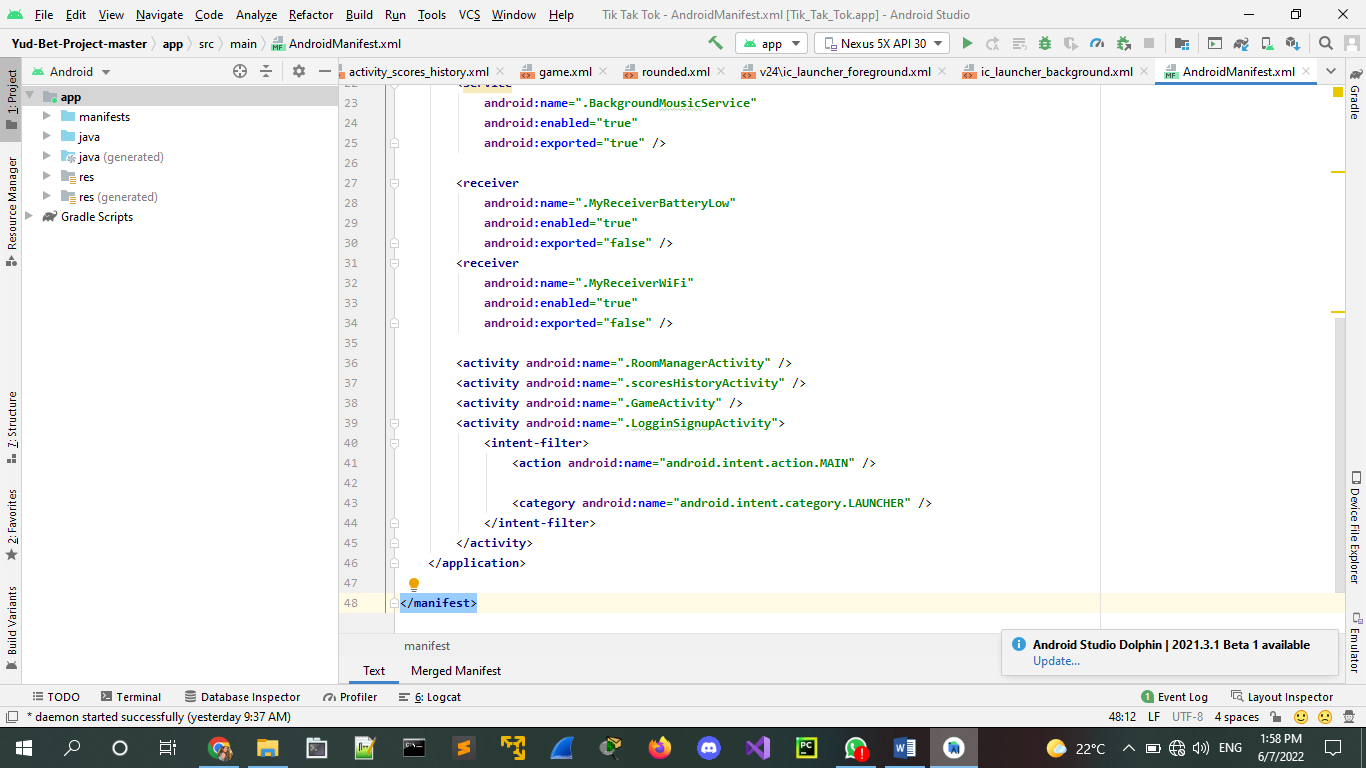
**אני מבקשת מהמתאם לקחת את המערך ולהציגו ב-listView.**

**#הערה:**

**יכולתי גם להשתמש בadapter בסיסי שהשפה נותנת לי אבל העדפתי לעצב בעצמי את הטבלה ולכן החלטתי לכתוב את ה adapter בעצמי.**

**8) קובץ המניפסט**

**מטרתו של קובץ זה לתאר את כל ה activities, Broadcast Receivers, services וההרשאות שקיימות אצלי בפרויקט.**

****

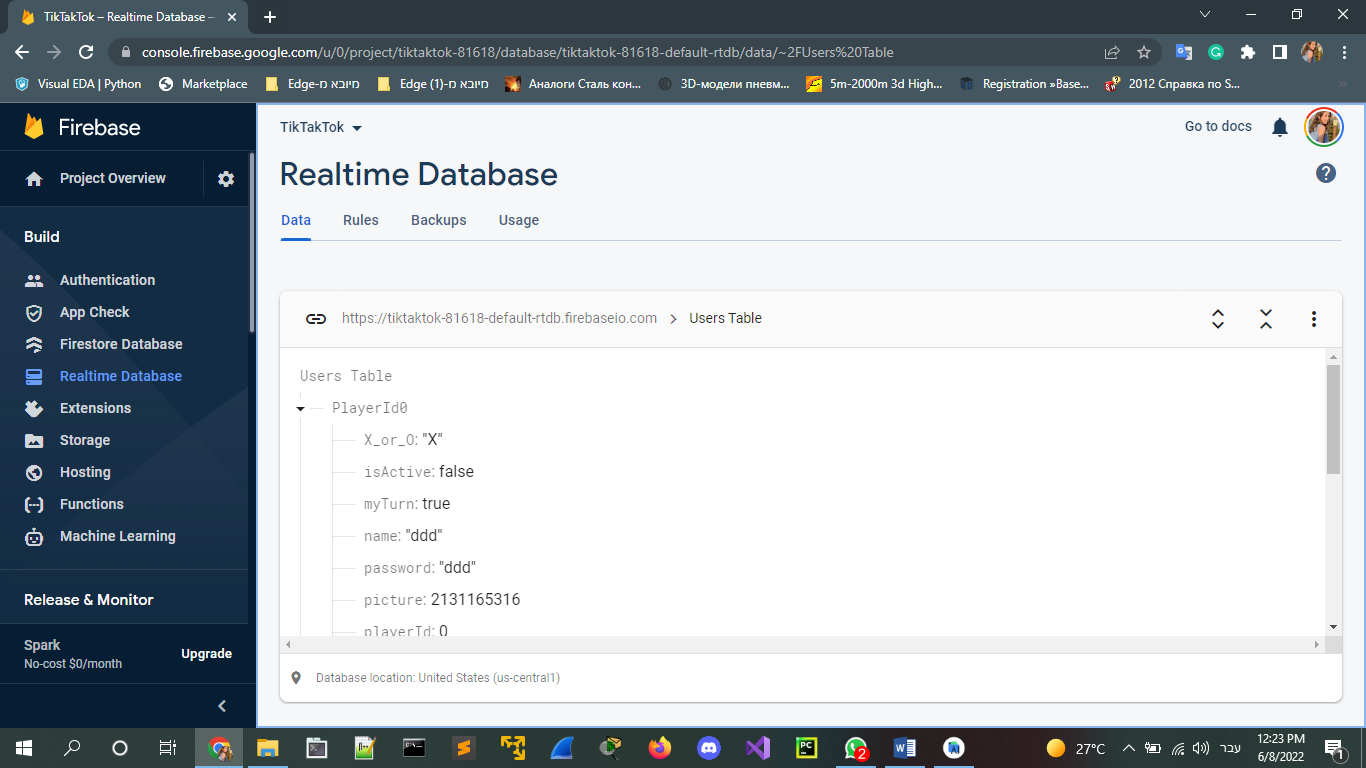
**11) הפיירבייס**

**הפיירבייס הוא שרת חינמי של גוגל שמאפשר לנו שמירת נתונים, תמונות וכו' בלי התעסקות בכתיבת קוד צד השרת. אני השתמשתי בפיירבייס כדי לאחסן את כל חדרי המשחק שלי וכדי שהשחקנים יוכלו לשחק זה עם זה ממכשירים שונים.**

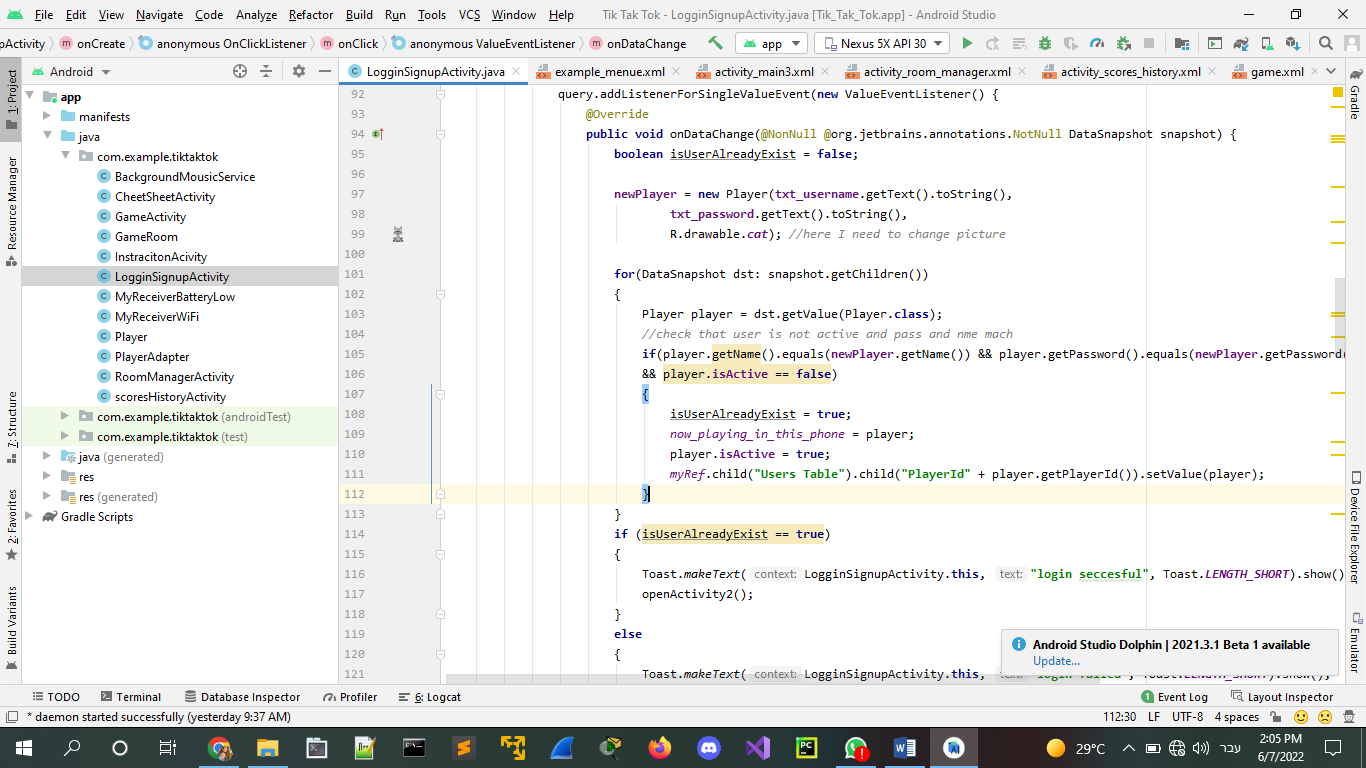
# מבנה הנתונים בפיירבייס

**בפרויקט השתמשתי בfirebase כמבנה נתונים שיכול לקשר בין** **השחקנים והחדרים.**

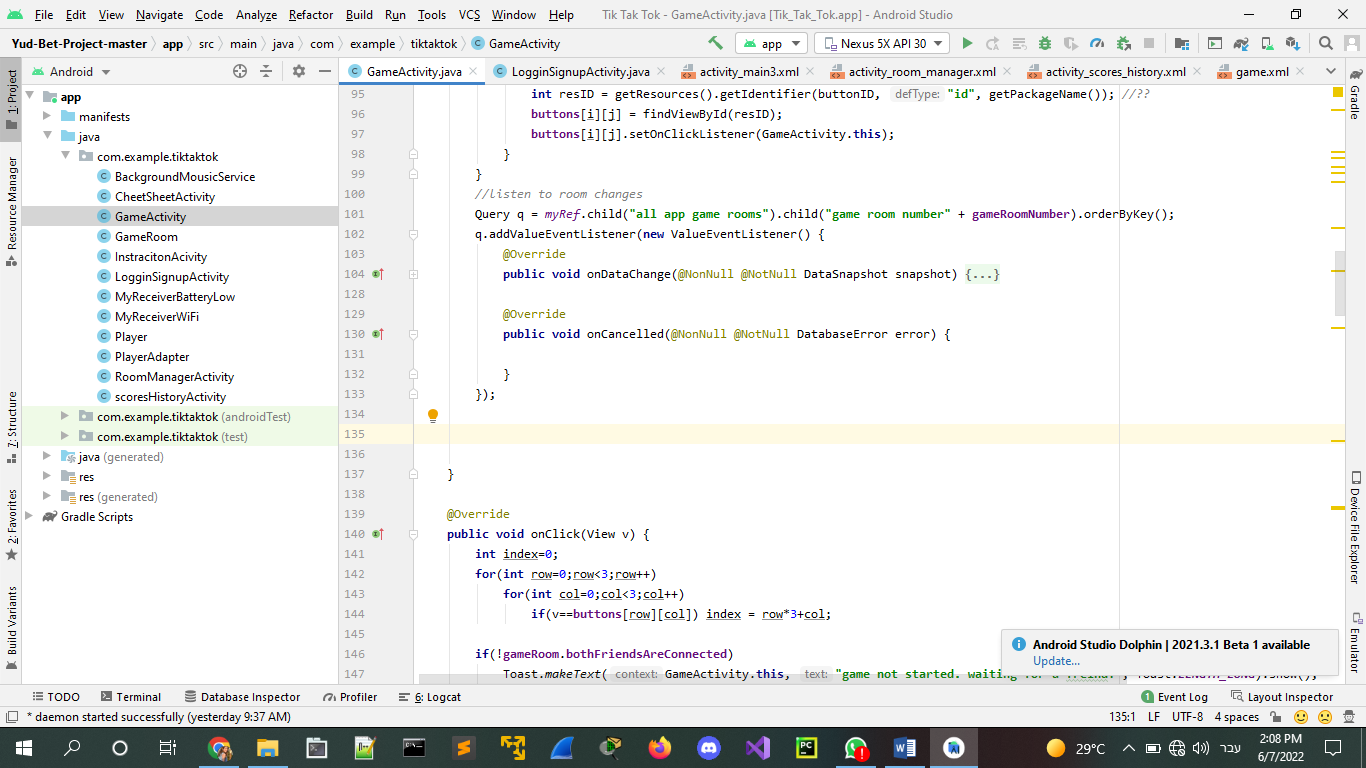
****

****

**דוגמה לקוד:**

****

**בקוד זה אני ניגש לפיצול ספציפי בפיירבייס שמתייחס לחדר משחק מסוים ואני מעדכן אותו בעזרת פקודת setValue.**

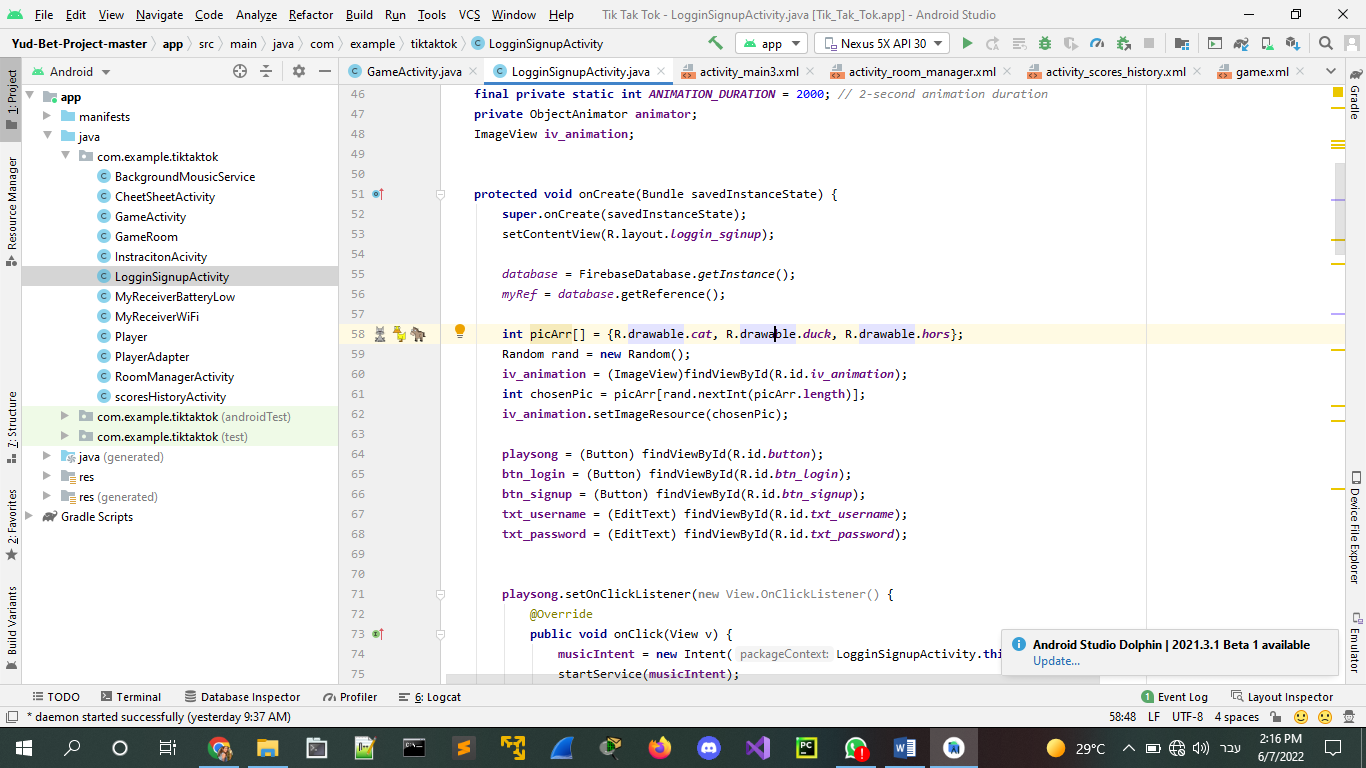
****

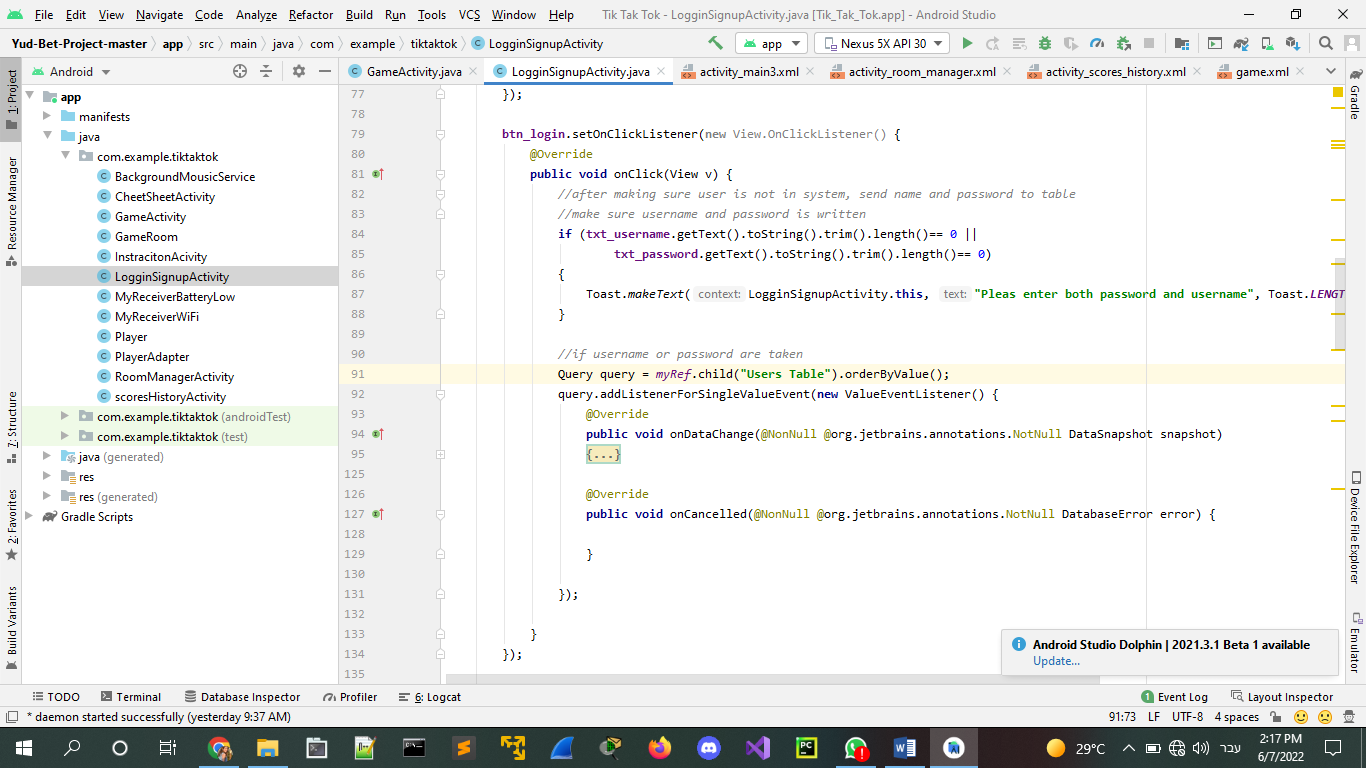
**בקוד זה אני מאזין לאירועים שמתרחשים בחדר מסוים.**

**דוגמה לאירועים בקוד שלי:**

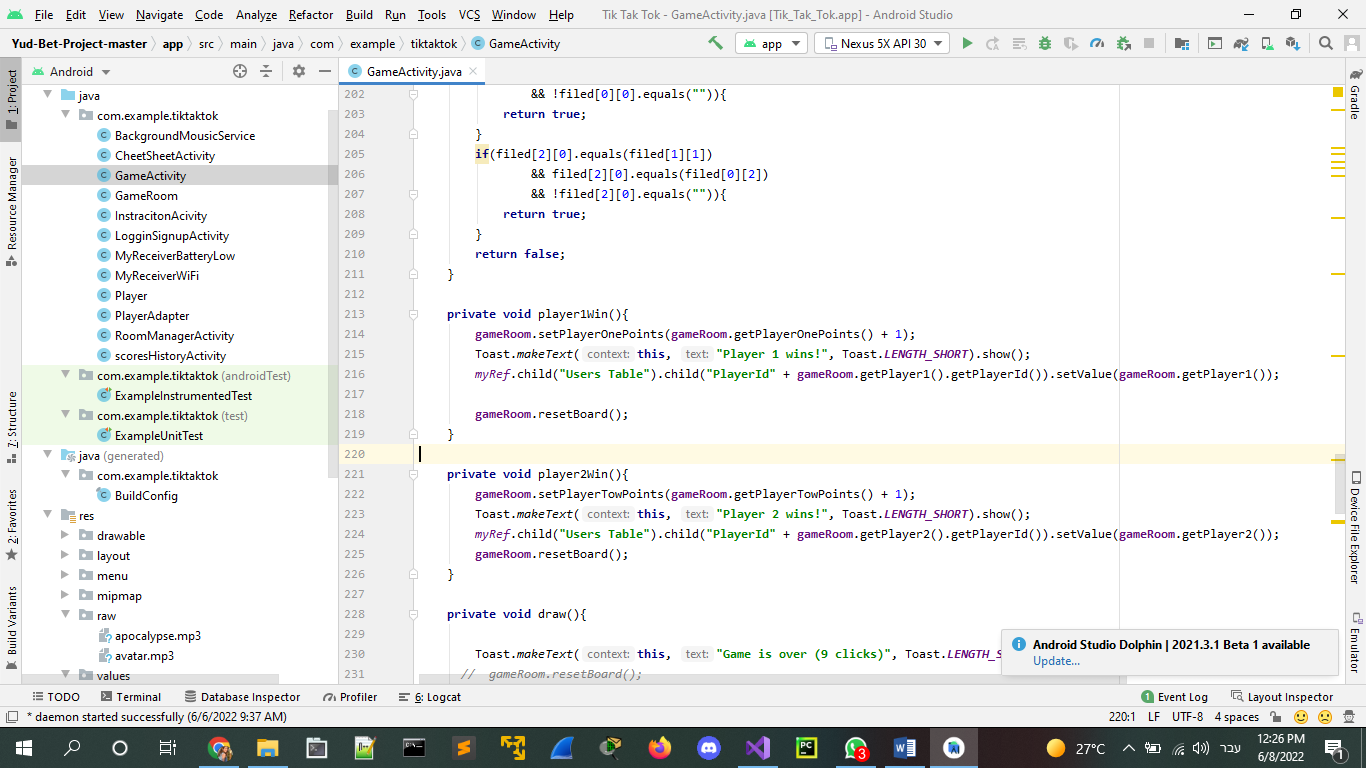
* **הצטרף שחקן חדש למשחק**
* **סימון משבצת**
* **ניצחון של אחד השחקנים**

**קוד לאחסון בפיירבייס:**

****

****

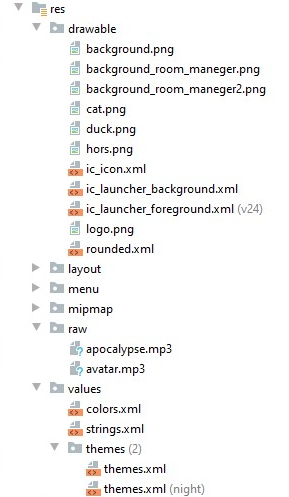
**בקוד הנ"ל ניתן לראות כי אני ניגש לאחסון בפיירבייס ובתוכו אני ניגש לטבלת המשתמשים ומקבל את המשתמשים הרשומים.**

**קישור בין הציורים שב-*storage* של פיירבייס לבין ה-*real-time-database* של פיירבייס:**

**בקוד הנ"ל ניתן לראות כיצד אני מקשר את הניקוד של השחקן שבאפליקציה בחדר המשחק ל real-time-database, בכך שאני ניגשת לטבלת שחקנים ומוצאת את השחקן המתאים לפי הID ומעדכנת את הניקוד שלו בעזרת פקודת setValue.**

**הנה דוגמה לאיך זה נראה ב- real-time-database   
**

# קבצים נלווים-תמונות, צבעים ומחרוזות

****

**תמונות של האייקונים באפליקציה**

**קובץ ווידאו ומוזיקה באפליקציה**

****

**מחרוזות באפליקציה**

**צבעים באפליקציה**

****

# סיכום

**ממש נהניתי לבנות את האפליקציה שלי, לפתח מוצר מתחילת הרעיון הראשוני ועד הביצוע והתוצר הסופי. הפרויקט הזה היה מאוד משמעותי עבורי משום שזאת פעם ראשונה שפיתחתי אשכרה אפליקציה משלי שעובדת ממש והכל אם זה לפתח רעיון ולתכנון, לעצב את המבנה הכללי של האפליקציה, לעצב את כל החלונות, ליצור מבנה נתונים מתאים ולכתוב את הקוד של כל החלקים.**

**אבל לא הכל ורוד בעולם התכנות ובפרויקט שלי התעוררה בעיה של חזרה אחורה , מעבר הפוך בין מסכים. כאשר ניסיתי לחזור אחורה הפרויקט היה נתקע, או שהמצג היה מהבהב וקורס. בעיה זו נגרמה מכייון שכאשר חזרתי למסך ההתחברות, ניתקתי את השחקן, ומחקתי אותו, ואז האפליקציה ניסתה לגשת לשחקן שלא היה קיים יותר. כדי לפתור את הבעיה חיברתי את השחקנים לפיירבייס, וכאשר יש מעבר בין מסכים אני מעדכנת את הפיירבייס במידה וצריך למחוק את השחקן ולעדכן את האפליקציה.**

**מהפרויקט הזה למדתי איך לתכנן אפליקציה ושיטות שונות ודרכים שונות לעצב חלונות. למדתי איך לתכנן מבנה נתונים עם כמה טבלאות והרבה משתנים שונים ואיך לעבוד איתם. ובעיקר קיבלתי מהפרויקט הזה הרבה תחושת מסוגלות וסיפוק מההצלחה והסיום של האתגר.**

# נספחים- רשימות תוכנה מלאות