A Deck of Cards



מגיש: ערן גודסי

ת.ז: 214495517

מנחה: ענת שלוס

ביה"ס: החקלאי אורט פרדס-חנה כרכור

תאריך הגשה: 31.05.2021

חלופה: אנדרואיד

0. תוכן העניינים:

[0. תוכן העניינים 1](#_Toc73314782)

[1. הקדמה ומבוא 2](#_Toc73314783)

[2. תיאור התוכנה ומטרותיה 2](#_Toc73314784)

[2.1. דרישות ומגבלות 2](#_Toc73314785)

[2.2. תרשים זרימת המסכים 3](#_Toc73314786)

[2.3. תיאור המסכים 4](#_Toc73314787)

[2.3.1 מסך ההתחברות – AuthActivity 4](#_Toc73314788)

[2.3.2 מסך רשימת החדרים – RoomListActivity 4](#_Toc73314789)

[2.3.3 מסך יצירת החדר – CreateRoomActivity 5](#_Toc73314790)

[**2.3.4** מסך המשחק – GameActivity 5](#_Toc73314791)

[3. מבנה התוכנה ומסד הנתונים 6](#_Toc73314792)

[3.1 מבנה התוכנה 6](#_Toc73314793)

[3.1.1 ההתממשקות ל-OpenGL 6](#_Toc73314794)

[3.1.1.1 העברת מידע הציור 6](#_Toc73314795)

[3.1.1.2 תכנות שיידרים 6](#_Toc73314796)

[3.1.1.3 ציור מקובץ (Batching) 7](#_Toc73314797)

[3.1.1.4 מחלקות המודול 8](#_Toc73314798)

[3.1.2 פירוט שאר המחלקות 9](#_Toc73314799)

[3.2 מבנה מסד הנתונים 11](#_Toc73314800)

[3.2.1 מבנה המשתמש 11](#_Toc73314801)

[3.2.3 מבנה החדר 11](#_Toc73314802)

[4. דרישות הפרויקט 13](#_Toc73314803)

[5. סיכום אישי 14](#_Toc73314804)

[6. ביבליוגרפיה 15](#_Toc73314805)

# 1. הקדמה ומבוא

במהלך הקורונה, האפשרות להיפגש עם אנשים אחרים נהייתה בלתי זמינה, ועל אף שעדיין ניתן לשמור על קשר דרך אפליקציות כמו Zoom ושיחות טלפון, פעילויות בין-אישיות רבות, ביניהן המשחק במשחקי קופסה, ובפרט במשחקי קלפים, נהיו בלתי זמינות. כדי להתמודד עם העובדה הזו, ניסיתי לחפש אחר אפליקציה שתספק את אותה החוויה של משחק קלפים משותף סביב שולחן, אך לא הצלחתי למצוא שום פתרון קיים שהיה נוח ונגיש לשימוש לטלפון.

הפרויקט שלי, ששמו הזמני הוא a Deck of Cards, מנסה לדמות את החוויה הזו, בצורה נוחה ואינטואיטיבית. האפליקציה נותנת למשתמשים חלל משותף עם קלפים בו הם יכולים לעשות כל מה שעולה על רוחם – לשחק בכל משחק שירצו, לארגן את השולחן כראות עיניהם, ואפילו לבלגן את כל השולחן כשמישהו מפסיד ורוצה בכך. הפרויקט מספק יכולת של יצירת "חדרים", אותם המשתמשים יכולים ליצור ואליהם להיכנס כדי לשחק יחדיו בקלפים.

האפליקציה לא מנהלת את המשחקים (כלומר, אי אפשר ליצור "חדר מלחמה" או "חדר רביעיות" בהם האפליקציה תנהל את החוקים והחלוקה), אלא נותנת למשתמשים בה שליטה מלאה על מה קורה בחדריהם.

# 2. תיאור התוכנה ומטרותיה

**מטרת התוכנה** היא לדמות את חוויית המשחק סביב שולחן קלפים, עם כל הפרטים הקטנים, בצורה נוחה ואינטואיטיבית.

**הפיצ'רים** שמספקת האפליקציה למשתמש הן כדלהלן:

* הרשמה והתחברות באמצעות מייל וסיסמא
* יצירת חדרי משחק עם שם, מגבלת אנשים, וסיסמא אופציונלית
* התחברות לחדרי משחק ואינטראקציה עם הקלפים הנמצאים בהם

## 2.1. דרישות ומגבלות

התוכנה דורשת תמיכה של הטלפון בOpenGL ES 3, מכיוון שהיא משתמשת בAPI של OpenGL כדי לצייר את הקלפים.



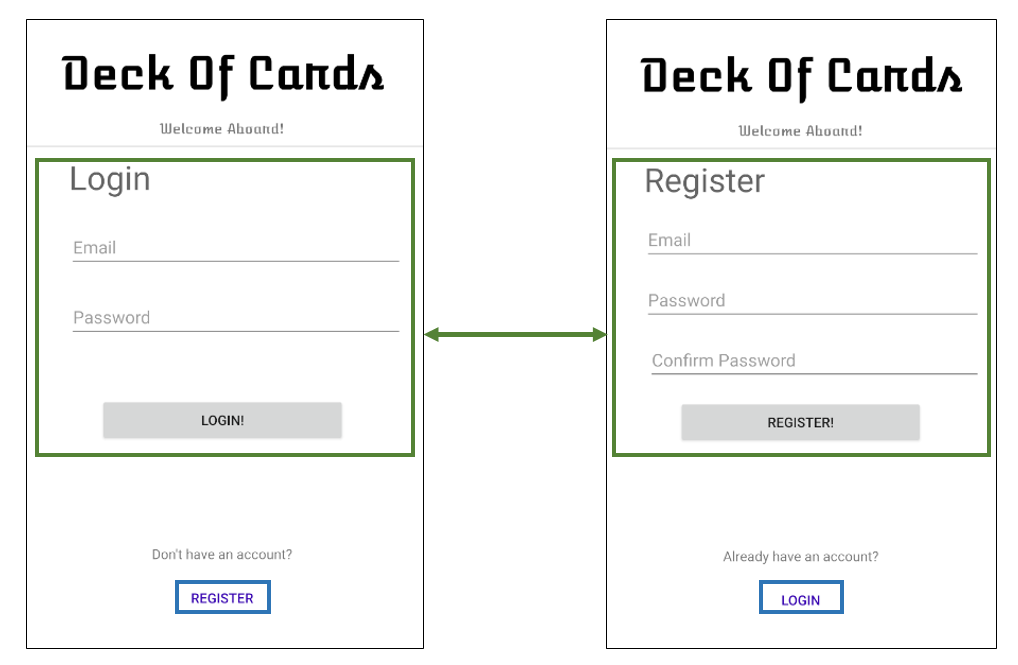
## 2.2. תרשים זרימת המסכים

## 2.3. תיאור המסכים

### 2.3.1 מסך ההתחברות – AuthActivity

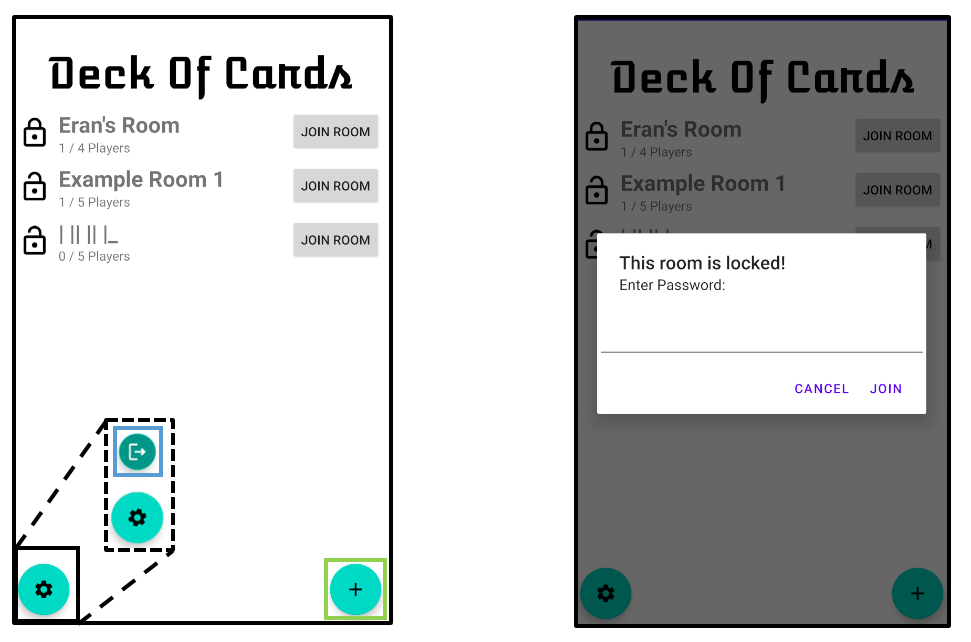
מסך ההתחברות מכיל בתוכו גם התחברות וגם הרשמה, כ2 Fragments שונים המוצגים באמצעות ViewPager. הכפתורים המסומנים בכחול (שהם למעשה אותו כפתור המשנה את תוכנו) מחליפים בין הFragments הלוך ושוב.

לחיצה על כפתור ההרשמה/התחברות מוודאת כי הפרטים תקינים, ואם כן יוצרת קשר עם Firebase. כאשר ישנה הצלחה, המשתמש מועבר למסך רשימת החדרים, וכאשר אין, מוצגת שגיאה למשתמש דרך שדות הטקסט.



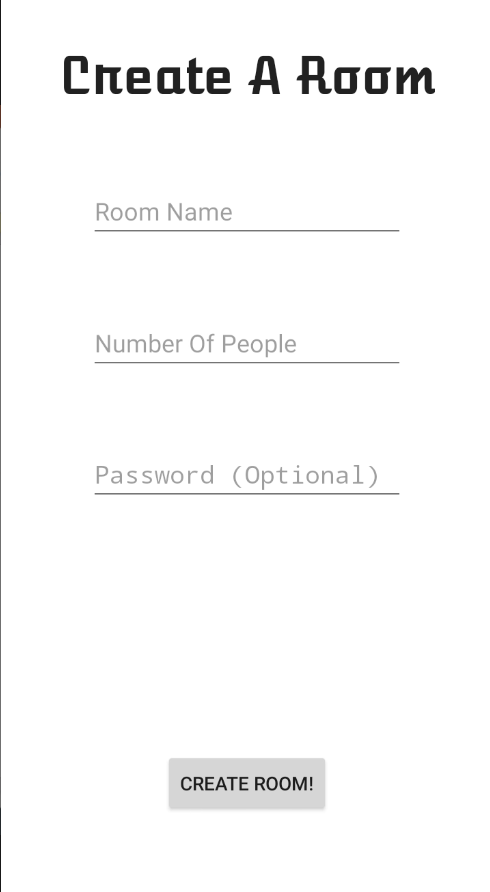
### 2.3.2 מסך רשימת החדרים – RoomListActivity

מסך רשימת החדרים מכיל בתוכו את כל החדרים ומאפשר התחברות לחדר נבחר. אם החדר נעול (מוגן בסיסמא)... ישנה אפשרות לגרור את האצבע כלפי מטה כדי לטעון מחדש את הרשימה. בנוסף המסך מכיל את האפשרות להתנתק מן האפליקציה ומאפשר מעבר למסך יצירת החדרים



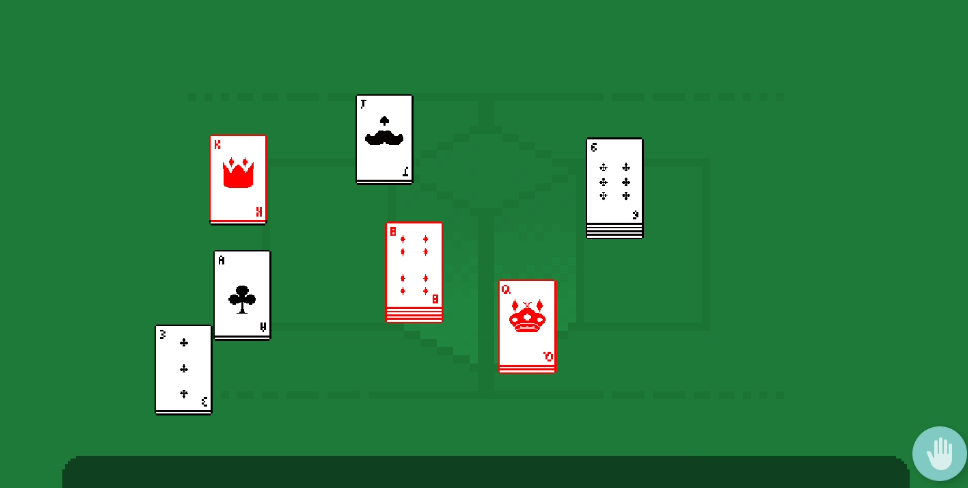
### 2.3.3 מסך יצירת החדר – CreateRoomActivity

מסך יצירת החדר נותן למשתמש להזין פרטים על החדר שברצונו ליצור. הפרטים הינם שם החדר (איך שיוצג ברשימה), כמות השחקנים המקסימלית בו וסיסמא (אופציונלית) לחדר. כאשר נלחץ כפתור יצירת החדר, נשלחת בקשה ל-Firebase כדי לוודא ששם החדר אינו תפוס. אם הוא תפוס תוצג הודעת שגיאה למשתמש, ואם לא הוא יוחזר למסך רשימת החדרים.



### **2.3.4** מסך המשחק – GameActivity

מסך זה הינו המסך בו מתרחשים המשחקים עצמם. המסך משתמש בOpenGL על מנת לצייר ביעילות בזמן אמת את מצב החדר אותו הוא מציג (באמצעות שימוש בGLSurfaceView). בנוסף על כך הוא מנהל תקשורת תמידית עם הFirebase כדי להשאר מסונכרן בנוגע למצב החדר ולעדכן אותו. המסך מאפשר למשתמש לבצע אינטראקציה עם הקלפים והערמות בהן הם מסודרים. על OpenGL, התקשורת עם Firebase, והאינטראקציה של המשתמש עם המסך אפרט בחלקים הבאים.



# 3. מבנה התוכנה ומסד הנתונים

## 3.1 מבנה התוכנה

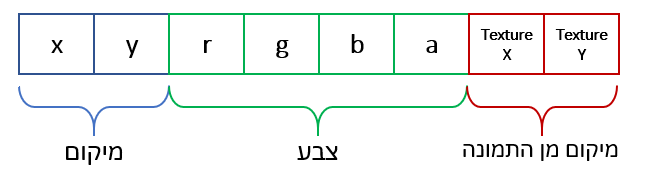
הקוד מחולק לשני חלקים עיקריים: חלק המתמקד בהתממשקות ל-OpenGL ומהלך המשחק, וחלק נוסף הקשור למסכי האפליקציה והלוגיקה שמאחוריהם. האפליקציה כולה נמצאת תחת החבילה com.redsponge.carddeck

### 3.1.1 ההתממשקות ל-OpenGL

כל קוד ההתממשקות נמצא תחת תת החבילה gl. הקוד עוסק ביצירת רכיב עליו קיים Context של OpenGL, טעינת משאבים (תמונות) דרך הממשק, חישובי מצלמה ומיקומי מסך, ושליחת פקודות ציור אל הממשק.

#### 3.1.1.1 העברת מידע הציור

כדי לצייר באמצעות OpenGL יש להעביר למערכת מערך של קודקודים (Vertices). קודקוד כולל בתוכו מיקום (x, y) אך אינו מוגבל רק למידע זה. באפליקציה זו קודקוד כולל בתוכו מידע על מיקום, צבע, ותמונה (טקסטורה) שיש לצייר עבורו.



נספח - דוגמא לקודקוד המועבר ל-OpenGL

היות והקודקודים מועברים לממשק של OpenGL כמערך בתים ארוך, יש להגדיר בנוסף הוראות כיצד לפרש את הקודקוד באמצעות פקודת glVertexAttribArray:



נספח - העברת מבנה הקודקוד ל-OpenGL



נספח - תיעוד הפעולה glVertexAttribArray

ניתן לראות בנספח 2 את התיאורים של של תכונותיו של הקודקוד. כך יועברו לשלב הבא בתהליך הציור:

#### 3.1.1.2 תכנות שיידרים

שיידר (Shader Program) הינה תוכנה הרצה על כרטיס המסך (או המערכות המוטבעות של OpenGL בטלפון). היא כתובה בשפת תכנות דמויית C הקרויה GLSL ומטרתה להדריך את המערכת אילו מניפולציות יש לבצע על הקודקודים, ואילו צבעים יש לצייר על המסך בסופו של דבר. ישנם 2 חלקים לתוכנת השיידר שיש לכתוב על מנת לקבוע את תהליך הציור: שיידר הקודקודים ושיידר הפיקסלים.

**שיידר הקודקודים (Vertex Shader)** אחראי על מניפולציות במרחב לקודקודים המועברים אליו – הוא לוקח את המיקום המועבר דרך הקודקוד ובאמצעות אלגברה לינארית (עם מטריצות המועברות מן הקוד) הופך את המיקומים למיקומים המוזזים ע"פ המצלמה והממופים בטווח [-1, 1], שהינו טווח הציור של OpenGL.

**שיידר הפיקסלים (Fragment Shader)** אחראי על הפלט הסופי – אילו צבעים יש לצייר בכל פיקסל. הוא עושה זאת באמצעות לקיחת צבעים מטקסטורה המועברת מבחוץ ע"פ המיקום על הטקסטורה המועבר לו דרך הקודקוד (Texture X, Texture Y).

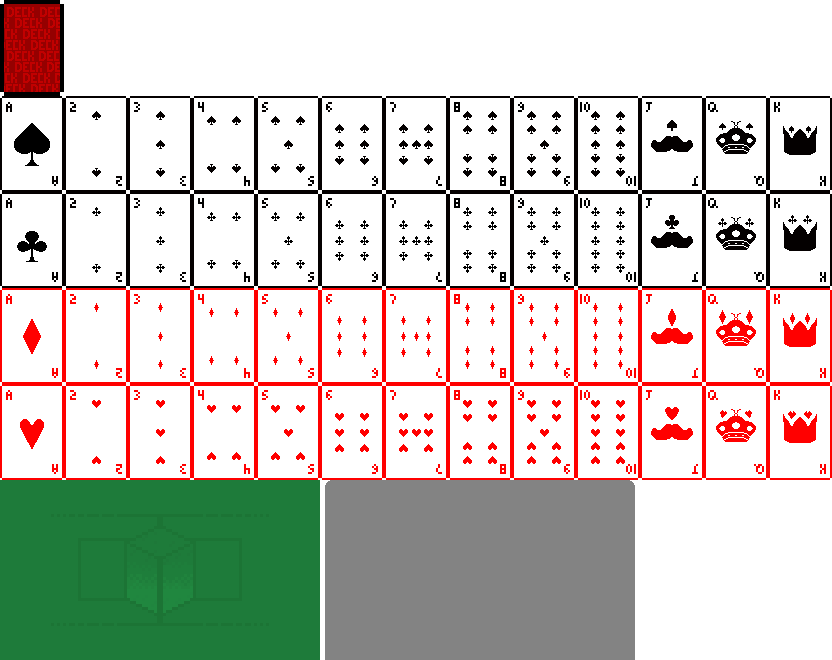


נספח - קוד שיידר הקודקודים והפיקסלים

#### 3.1.1.3 ציור מקובץ (Batching)

העברת המידע מן האפליקציה למערכת ה-OpenGL הינה צוואר הבקבוק המשמעותי בציור. על מנת שיתאפשר ציור מהיר של יותר מצורה או שתיים, יש לממש קיבוץ של צורות רבות לקריאת ציור אחת.

תהליך הקיבוץ מתממש בכך שבמקום לשלוח את הקודקודים ל-OpenGL ברגע שאני רוצה לצייר אותם, אני ממלא מערך float-ים בגודל גדול מאוד (1024) במידע על הקודקודים אחד אחרי השני. לאחר מכן, כאשר אני רוצה לצייר, אני שולח את כל הקודקודים בבת אחת (פשוט מעביר את המערך) יחד עם טקסטורה אחת גדולה הכוללת את כל הטקסטורות שארצה לצייר. כך, בקריאת ציור אחת, אני יכול לצייר מסך שלם.



נספח - טקסטורה גדולה הכוללת הכל

#### 3.1.1.4 מחלקות המודול

|  |  |
| --- | --- |
| GLGameView | המחלקה המקשרת בין OpenGL לבין ממשק האנדרואיד. מורישה מ-GLSurfaceView ומנהלת GLRenderer בתוכה שאחראי על הציור. בנוסף על כך מחלקה זו מאזינה לאירועי קלט מהמשתמש (לחיצה על המסך) ומעבירה אותן למאזין מטיפוס InputHandler. |
| GLRenderer | מחלקה המממשת את ממשק הGLSurfaceView.Renderer - היא המחלקה בה נמצאת פעולת הציור הראשית, והיא אחראית על תזמון והחלפת מסכי-ציור (עצמים מטיפוס Screen, עליו יפורט בהמשך) |
| RawReader | מחלקת עזר הקוראת קבצי טקסט השמורים בתיקיית raw ומחזירה את תוכנם כמחרוזות. |
| ShaderProgram | מחלקה העוטפת את מערכת השיידרים של OpenGL. היא מתקשרת עם הממשק על מנת ליצור ולקשר שיידרים על פי קוד שיידרים המועבר לה. בנוסף המחלקה מאפשרת העברת קבועים (Uniforms) לשיידר אותו טענה באמצעות פעולות עזר. |
| TextureBatch | המחלקה האחראית על הציור המקובץ של קודקודים. |
| Vector2 | מחלקה המייצגת ווקטור דו-ממדי, כלומר נקודה על המסך. מחזיקה ערך x ו-y ומספקת פעולות מתמטיות על הווקטור |
| BufferUtils | מחלקת עזר להקצאת מערכים מן החבילה java.nio – דרושים להעברת מידע לOpenGL |
| Disposable | ממשק המייצג מחלקה שיש לשחרר ממנה זיכרון. |
| Texture | מחלקה שעוטפת את מערכת הטקסטורות של OpenGL – טוענת ומשחררת טקסטורות. |
| TextureRegion | מחלקה המייצגת חלק מטקסטורה – מחזיקה טקסטורה ואת הנקודות המייצגות את המיקום של החלק מן הטקסטורה. |
| Camera | מחלקה האחראית חישוב מטריצות מיקום וProjection. מייצגת מצלמה בעלת מיקום דו ממדי. |
| Viewport | מחלקה מופשטת המתארת צג - היא מגדירה גודל למסך שאינו תלוי בפיקסלים על מנת לאפשר ציור זהה על מסכים שונים. המחלקה מאפשרת למחלקות אחרות לממש את פעולת הapply שלה. |
| FitViewport | מחלקה מממשת של Viewport: שומרת על הגודל המוגדר בכך ששומרת על היחס ומוסיפה Black bars (בכל צבע נבחר, לא רק שחור) בחלקים המתים של המסך |
| ScaleViewport | מחלקה מממשת של Viewport: שומרת על הגודל המוגדר בכך שמותחת את הציור. |
| InputHandler | ממשק המגדיר פעולות המטפלות בTouchScreen Input.  (onTouch, onRelease, onDrag) |
| Screen | מחלקה מופשטת המייצגת מסך על קונטקסט ה-OpenGL. כל מסך מעדכן את עצמו ומגיב לאירועים של האפליקציה (שינוי גודל, הצגה ראשונית, שחרור, הסתרה...) |
| GameScreen | מחלקה המממשת את הממשק Screen ומנהלת את המשחק |

למדתי כיצד לעבוד עם OpenGL מהאתר <https://learnopengl.com>

### 3.1.2 פירוט שאר המחלקות

|  |  |
| --- | --- |
| AuthOption | מתארת אפשרות לAuthActivity (הרשמה או התחברות) |
| AuthPagerAdapter | מתאם לPager שמעביר בין Fragment הרשמה והתחברות |
| AuthUtils | מחלקת עזר הבודקת דברים כגון האם המידע שהוזן חוקי (מגבלות על שם המשתמש והסיסמא) |
| AuthLoginFragment | פרגמנט המוצג במסך ההתחברות ואחראי על התחברות למשתמשים חוזרים |
| AuthRegisterFragment | פרגמנט המוצג במסך ההתחברות ואחראי על הרשמה של משתמשים חדשים |
| LoggedUser | מחלקה המייצגת משתמש מחובר – כוללת את שמו, ה-id שלו והמייל שלו. |
| CardData | מתארת את התצוגה של קלף על משחק המשחק – סוגו והאם הוא הפוך. |
| RoomFBC | מקבל Reference לחדר ומנהל אותו – קורא ומארגן את הקלפים, החבילות, וסדר החבילות של החדר. |
| PackedTextures | מפרק הטקסטורה הגדולה לTextureRegions שונים. אחראי על ייבוא הטקסטורה וסיפוק תתי-טקסטורות לכל סוגי הקלפים, הרקעים, ושאר התמונות. |
| Constants | אוחז בקבועים |
| PileData | מחלקה המתארת את התצוגה של חבילה על המסך – מיקום, גודל וזמן בחירה. זהו חלק ה-transform של כל חבילה ב-Firebase. |
| PileFBC | מחלקה המחברת בין Reference לחבילה בFirebase לבין הקוד – שומרת על סנכרון בין האפליקציה ומסד הנתונים, לרבות רשימת הקלפים והtransform |
| RoomFBC | מחלקה המחברת בין Reference לחדר בFirebase לבין הקוד, לרבות טעינת הקלפים, ניהול החבילות, ושמירת ועדכון סדר הציור שלהן. המחלקה מממשת את הממשק Iterable ועל כן מאפשרת לעבור על כל החבילות שבתוכה ע"פ סדר הציור |
| ListRoomItem | מחלקה פשוטה המייצגת חדר ברשימת החדרים, בלי הסנכרון המלא עם Firebase. |
| RoomAdapter | אדפטר לרשימת החדרים במסך רשימת החדרים |
| ChildEventAdapter | מחלקת עזר המממשת את ממשק הChildEventListener ומגדירה את פעולותיו כריקות |
| Listeners | מחלקת עזר לייצור מאזינים לFirebase באמצעות lambdas |
| MathUtils | מחלקת עזר עם פעולות מתמטיות |
| Pair | מחלקת עזר המחזיקה 2 עצמים |
| SynchronizedList | מחלקה המייצגת רשימה המסונכרנת עם רשימה ב-Firebase. רעיון המימוש מגיע מן האתר <https://firebase.googleblog.com/2014/05/handling-synchronized-arrays-with-real.html> המתאר עבודה עם מערכים מסונכרנים. |
| Utils | מחלקת עזר כוללת לשלל פקודות עזר מכל מיני תחומים. |

## 3.2 מבנה מסד הנתונים

הפרויקט משתמש ב-Firebase בתור מסד הנתונים שלי – ספציפית, במערכת הRealtime Database. המבנה הכללי של מסד הנתונים מורכב מרשימת משתמשים ורשימת חדרים

### 3.2.1 מבנה המשתמש

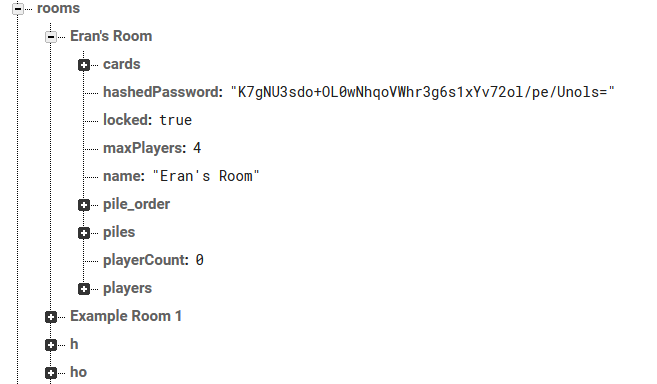
. כל משתמש נראה כך:



נספח - מבנה המשתמש בDatabase

הבחירה לשמור את הUID גם בתור המפתח וגם בתור תכונה נעשתה מתוך נוחיות בפרסור של המידע בקוד (שדה הUID נכנס למשתנה UID בתוך עצם LoggedUser)

### 3.2.3 מבנה החדר

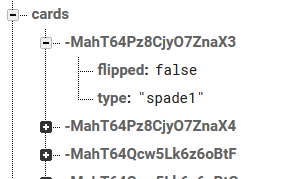


נספח - דוגמא לחדר

החדר מורכב מ2 חלקים – חלק הרשימה, וחלק המשחק. בחלק הרשימה נכללים שם החדר , כמות השחקנים המקסימלית בחדר, וסיסמתו (לאחר העברה בSHA256 והמרה לבסיס 64) כל אלו משמשים את האפליקציה במהלך מסך רשימת החדרים – הם מתארים כיצד להציג חדר והאם לנעול אותו.

חלק המשחק כולל את רשימת הקלפים, רשימת החבילות, וסדר החבילות. המשחק כולו מאורגן בחבילות הכוללות קלף 1 או יותר, אותן אפשר להזיז.

רשימת הקלפים כוללת את כל הקלפים שהולכים להופיע במשחק (בתוך חבילות). כל קלף מתואר באמצעות סוגו (טיפוס ומספר) והאם הוא הפוך. המפתח של קלף משמש להתייחסות אליו בחבילות.



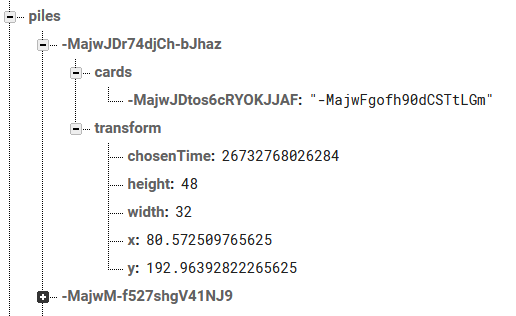
נספח - רשימת הקלפים בחדר

סדר החבילות הינו מערך סינכרוני המחזיק את סדר הציור של החבילות בFirebase – הוא מעודכן בתכיפות ע"י האפליקציות השונות ולמעשה הינו המערך הערוך ביותר (משתנה בכל רגע מכיוון שכל לחיצה על חבילה מעבירה אותה לתחילת המערך).



נספח - סדר החבילות

כל חבילה מורכבת מ2 חלקים: חלק הtransform וחלק הcards. הtransform של חבילה כולל את המיקום שלה והאם היא נבחרה (כזמן הבחירה כדי להמנע מסיבוכים בהחזקת בוליאן). הcards כולל את רשימת הקלפים בחבילה כמערך סינכרוני.



נספח - פירוט חבילה

# 4. דרישות הפרויקט

* Activity – באפליקציה נעשה שימוש ב4 מחלקות Activity המייצגות את המסכים השונים
* Intent – באפליקציה נעשה שימוש בIntent-ים על מנת להעביר מידע בין מסכים ולפתוח מסכים שונים.



* Service – באפליקציה נעשה שימוש בService של חיישני הטלפון, וזאת כדי לאפשר ששקשוק של הטלפון יפזר את קלפי המשחק.  
  

# 5. סיכום אישי

הפרויקט שלי החל מהרצון לשחק קלפים דרך הרשת, אבל יותר מזאת, מהרצון לתכנת משחק. כתיבת גרפיקה ומשחקים הם הדברים שאני הכי מתחבר אליהם בתכנות, ובגלל זה הרעיון של הפרויקט משך אותי מאוד.

במהלך הפרויקט למדתי לעבוד עם המגבלות של OpenGL במערכות מוטמעות (OpenGL ES) – הייתי צריך לממש אלגוריתמים לייעול התוכנית ולהגביל את כמות המשאבים שהשתמשתי בהם (לדוגמא – כיווץ הכל לטקסטורה אחת). הדבר היווה אתגר אבל מאוד נהניתי ממנו.

הנקודה שהיתה עבורי הכי קשה היתה ככל הנראה העבודה והייעול של Firebase בקונטקסט אסינכרוני. הייתי צריך לתאם בין קריאות, כתיבות, וציור תמידי אל המסך. בעקבות זאת הייתי צריך לתכנן כיצד לתכנת סביב האסינכרוניות, ולשמור על דברים מתואמים בין גרסאות. הקוד דרש איטרציות רבות, אך אני מרגיש שזה אחד הדברים בהם הכי התפתחתי במהלך הפרויקט. אני בטוח שאשתמש בFirebase בעתיד, בעיקר כי כעת אני הרבה יותר מיומן במערכת.

בסך הכל, מאוד נהניתי מתכנות הפרויקט – למדתי רבות, ואני מרוצה מהתוצר הסופי עד מאוד.

# 6. ביבליוגרפיה

מידע על יצירת קונטקסט של OpenGL באנדרואיד – https://developer.android.com/guide/topics/graphics/opengl

מדריך דרכו למדתי איך לעבוד עם OpenGL – https://learnopengl.com

מדריך המפרט אלגוריתם לרשימות מסונכרות בFirebase -   
https://firebase.googleblog.com/2014/05/handling-synchronized-arrays-with-real.html