

Chess champion נושא הפרויקט: משחק



תוכן העניינים

3.....	מבוא
6.....	מדריך למשתמש
7.....	תרשים זרימה בין מסכים
8.....	תכנון ותיעוד מסכי הפרויקט
25.....	תיאור תחום הידע
35.....	תרשים מחלקות (UML)
38.....	מחלקות הפרויקט
45.....	בסיס נתונים מרוחק (Firebase)
47.....	בסיס נתונים מסוג Firestore
49.....	SharedPreferences
50.....	Services
51.....	Camera & Gallery
53.....	רכיבים מתקדמים נוספים
55.....	ביבליוגרפיה



מבוא

תיאור

שם הפרויקט, תיאור קצר של הפרויקט (התיאור אמור להסביר לקורא מה מתרחש בפרויקט).
 "Chess Champion" ("אלוף השחמט") היא אפליקציית שחמט ייחודית המשלב את המשחק הקלאסי עם חידושים הבאים לידי ביטוי בעיצוב המשחק וביכולת לבחור בעצמך את חוקי המשחק.

שחמט רגיל:

האפליקציה מאפשר לשחק את המשחק בחוקי המסורתיים אך עם שיפורים מודרניים, כגון שעון עצר מובנה, הוספת והורדת זמן במהלך המשחק וסימון הצעדים החוקיים של כל חייל.

שחמט מותאם אישית:

האפליקציה דואגת לתת כמה שיותר חופש למשתמש בעיצוביות של המשחק. המשתמש יכול לבחור את צבעי המשבצות המרכיבים את הלוח ואת התמונות המייצגות את החיילים על הלוח – בעזרת תמונות מהגלריה או בעזרת צילום תמונות חדשות.

אסתטיקה מותאמת אישית:

האפליקציה מקדשת את חירותו של המשתמש ונותנת לו את האופציה לשלוט בעיצוב ובחוקי המשחק. גולת הכותרת של האפליקציה היא מצב המשחק Custom Chess. במצב משחק זה המשתמש ראשית בוחר את גודל הלוח (משבצותXמשבצות) ומיקום החיילים ההתחלתי, לאחר מכן בוחר המשתמש את התנועות והמתקפות החוקיות של כל סוג חייל שמופיע על הלוח ולבסוף מתחיל המשתמש את המשחק. אופציה זו מעניקה למשתמש את היכולת להתאמן במצבי שחמט שונים ולהביע את היצירתיות שלו על ידי בחירת חוקים חדשים ומקוריים ולאתגר את חבריו.

קהל היעד

למי מיועד המשחק? לאילו אוכלוסיות? גילאים?

Chess Champion מיועד למגוון רחב של אנשים, ומותאם לאוכלוסיות וקבוצות גיל שונות. המשחק פונה לקהלים רחבים כגון:



- חובבי שחמט ואסטרטגיה: המשחק מכיל את משחק השחמט הקלאסי ובנוסף נותן למשתמשים לחקור ומצבים שונים במשחק. כמו כן, המשחק מאפשר גם למיומנים ביותר להתנסות בגירסא שונה ומרעננת של המשחק.
- אוהבי עיצוב ויצירה: אוהבי עיצוב ישאבו למשחק וישתמשו ביצירתיות שלהם כדי ליצור לוחות בצבעים שונים, חיילים מעוצבים וחוקים יצירתיים.



רקע

הסיבות לבחירת הנושא, איך ולמה הגעתם דווקא לנושא הזה.

בתור חובב שחמט מגיל צעיר תמיד נמשכתי למשחק והתפעלתי מהתחכום שלו, במיוחד אהבתי לחפש סוגי משחקי שחמט עם חוקים אחרים (כגון שח הפוך, שח אטומי, שח שוודי ועוד). כאשר התבקשתי לבחור נושא לפרויקט הגמר שלי רציתי לבנות אפליקציה שאשתמש בה, כמו כן ידעתי שארצה לבנות משחק משום שהוא מפיק שעשוע ויצירתיות. לאחר חשיבה ממושכת החלטתי שאני רוצה לבנות משחק שחמט בגלל שידעתי שיצירתו תהווה אתגר אך תהיה משתלמת – מבחינת הפיתוח האישי שלי כמתכנת ומבחינת איכות הפרויקט.

הצגתי את הרעיון למורה שלי והוא הסביר לי ששחמט הוא אינו רעיון מקורי לאפליקציה ועלי להביא חידוש שיגרום לייחודיות הפרויקט. אני חשבתי על שילוב אספקט העיצוב באפליקציה והמורה הציע לי להוסיף את מצב ה Custom Chess. בהתחלה חשבתי שלתכנת את המצב השחמט המיוחד הוא קשה מאוד על סף הבלתי אפשרי, אבל ככל שהתקדמתי בפרויקט קיבלתי יותר בטחון בחומר וביכולת שלי לסיים משימות.



מדריך למשתמש

פרק זה יכיל הוראות התקנה – הסבר לדרישות ההתקנה בסביבת העבודה.
ההסבר מיועד עבור המשתמש ולכן צריך להיות פשוט וברור, מבלי להיכנס לפרטים טכניים.

גרסת אנדרואיד מינימלית

API level 21, Android 5

מכשירים עליהם נבדקה המערכת

אימולטור - Nexus 4 Android 11

הרשאות

- הרשאה לקריאה מהאחסון. נדרשת לצורך קריאת התמונות של המשתמש.
- הרשאה לאינטרנט. נדרשת לצורך שמירת ושליפת נתונים מהענן.

דרישות מיוחדות ואילוצים

- מגבלות או אילוצים באיזשהם אלמנטים בחלון / מסך (מס תווים, גודל קובץ, גודל תמונה, וכו')
- האפליקציה חייבת WIFI במכשיר.

אופן הפעולה של הפרויקט

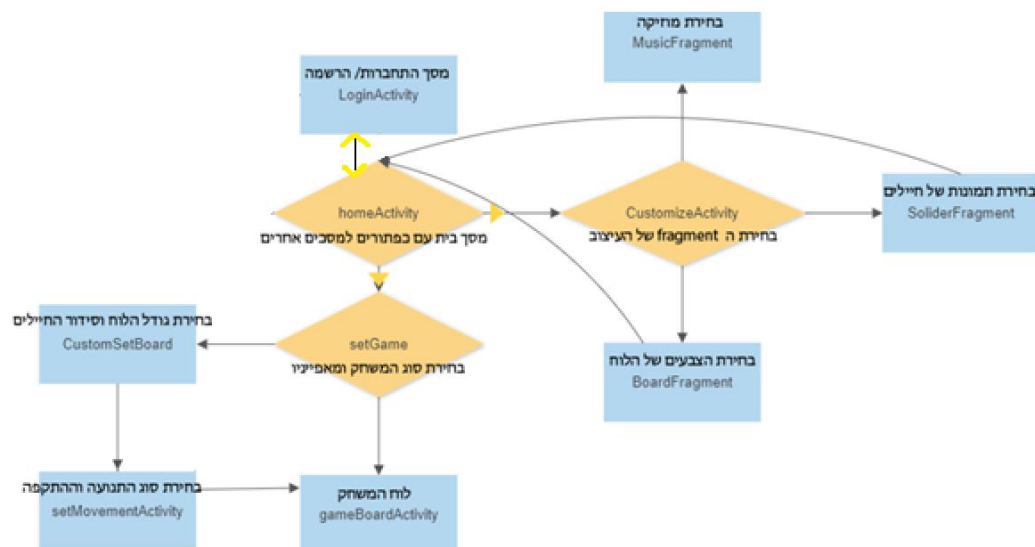
תיאור כללי של האפליקציה ללא כניסה לחלקים טכניים. המטרה היא להסביר למשתמש איך להשתמש באפליקציה ומה היכולות הקיימות בה.

בכניסתך הראשונית לאפליקציה יופיע מסך ההתחברות. במידה ועדיין לא יצרת משתמש באפליקציה, עליך ללחוץ על כפתור הרישום וליצור משתמש חדש. לאחר ההתחברות תועבר למסך התפריט הראשי שם תוכל להתחיל מיד לשחק על ידי לחיצה על המלכה ואכילת הפיון העומד תחת המסך הרצוי...



תרשים זרימה בין מסכים

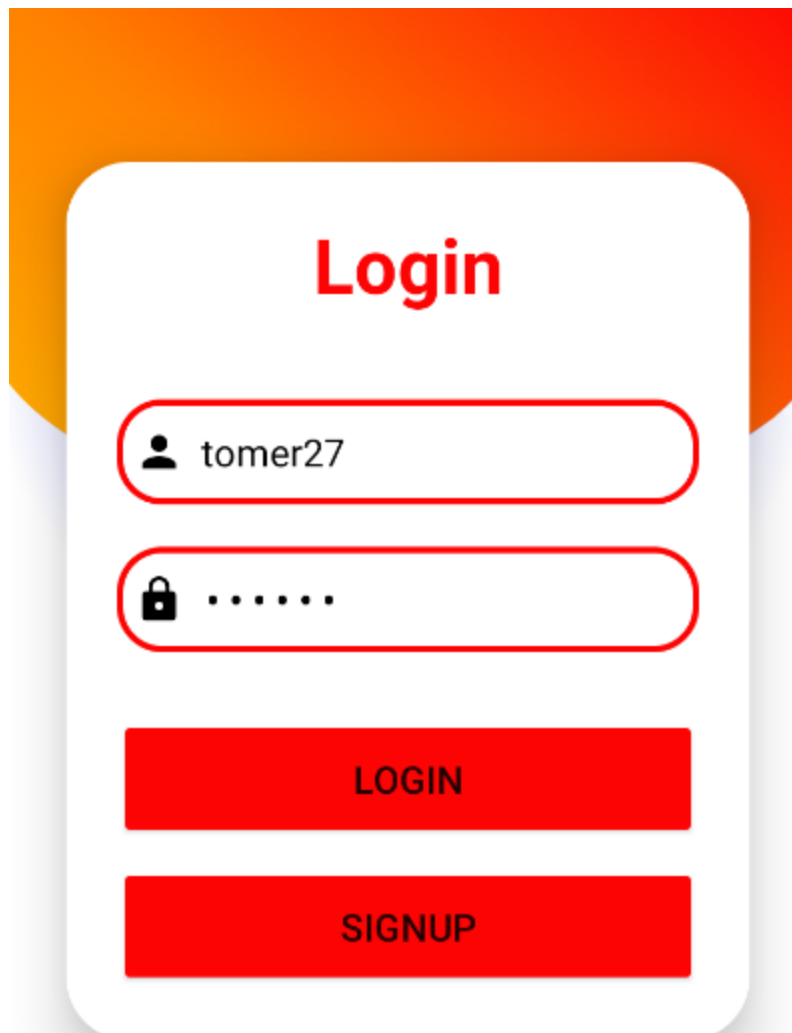
תרשים מסכים המתאר את היררכיית המסכים והמעברים ביניהם (Screen flow diagram).
 עבור כל מסך יש לכתוב את תפקידו בעברית ואת שמו של ה-Activity המייצג אותו. יש להציג מעבר אפשרי בין מסך למסך באמצעות קוים וחצים. אם מעבר אפשרי לשני הכיוונים, יש להציג חץ דו כיווני.




תכנון ותיעוד מסכי הפרויקט**מסך הפתיחה**


מה הוא מכיל והאם משמש נקודת ניווט.

בודק את השם משתמש והסיסמה, במקרה ומתאימה מכניס את המשתמש. בעת לחיצה על כפתור ההרשמה למטה, שם משתמש וסיסמה שהוכנסו ישמרו במערכת (אם השם משתמש לא קיים כבר) והמשתמש יוכנס למסך הבית.

A login screen mockup with a white rounded rectangle centered on an orange-to-red gradient background. The word "Login" is displayed in large red font at the top. Below it are two input fields with red borders: the first contains a person icon and the text "tomer27", and the second contains a lock icon and seven dots. At the bottom are two red rectangular buttons with white text: "LOGIN" and "SIGNUP".

Login

 tomer27



LOGIN

SIGNUP



עבור כל שאר המסכים (דוגמאות החל מעמוד 7):

בחלק זה נפרט את כל המסכים שיש לנו באפליקציה. עבור כל מסך יש לציין את הפרטים הבאים:

כותרת / שם / תפקיד המסך (מסך כניסה, מסך משחק, מסך רישום, מסך הוספת פריט וכו')

תיאור המסך (איזה מידע מציג, מה ניתן לבצע)

תמונת מסך

פירוט אלמנטים (עבור כל אלמנט תצוגה כדוגמת כפתור, תיבת טקסט, יש להסביר את תפקידם ואת האירועים שמתרחשים בעקבות הפעלתם).

מסך הבית

• מסך ניווט ראשי, ממנו נוח להגיע למסכים אחרים.

game יעביר אפליקציה

המשחק.

logout ינתק את

ויעביר אותו למסך

customize יעביר

למסך העיצוב.

והמשכת המוזיקה.



• כפתור set

למסך הגדרת

• כפתור

המשתמש

ההתחברות.

• כפתור

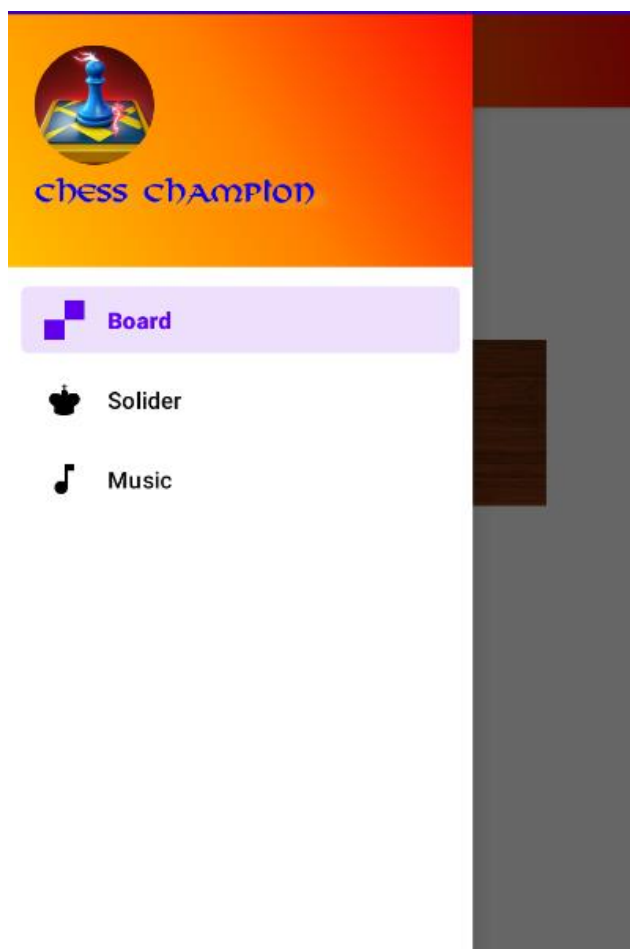
אפליקציה

• כפתור לעצירת



מסך customize

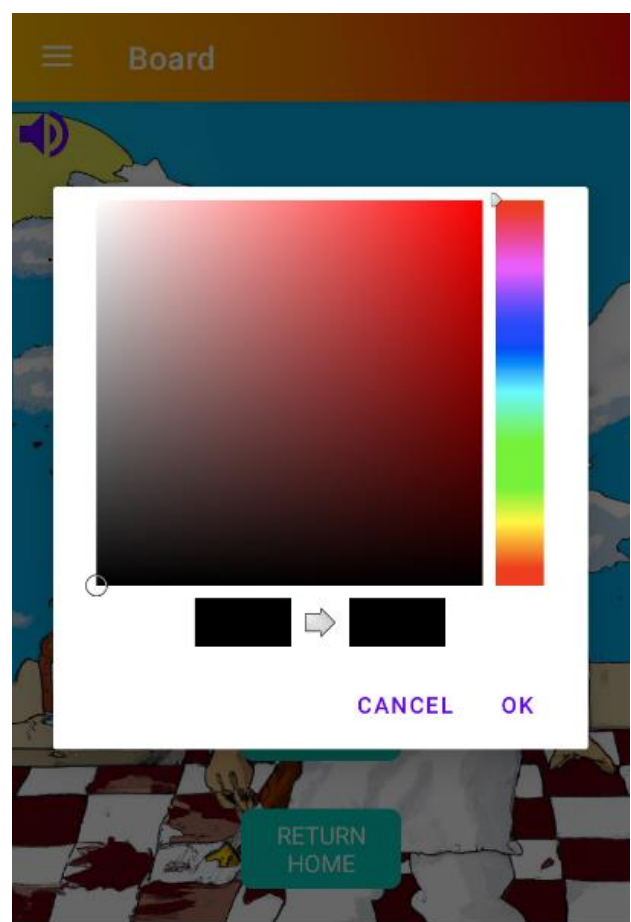
- כפתור Board יעביר אפליקציה לפרגמנט בו יבחרו צבעי הלוח.
- כפתור Soldier לעביר אפליקציה לפרגמנט בו יבחרו תמונות החיילים.
- כפתור Music יעביר אפליקציה לפרגמנט בו תבחר המוזיקה.



מסך Board

מסך זה הינו Fragment

- המשבצת הימנית תעביר אפליקציה לcolor picker בוא יבחר המשתמש צבע א'.
- המשבצת השמאלית תעביר אפליקציה לcolor picker בוא יבחר המשתמש צבע ב'.
- כפתור set color ישמור את ערכי הצבעים בפייר בייס.
- כפתור Return home יעביר את האפליקציה למסך הבית ויעדכן אותו לצבעי הלוח החדשים.
- כפתור לעצירת והמשכת המוזיקה.

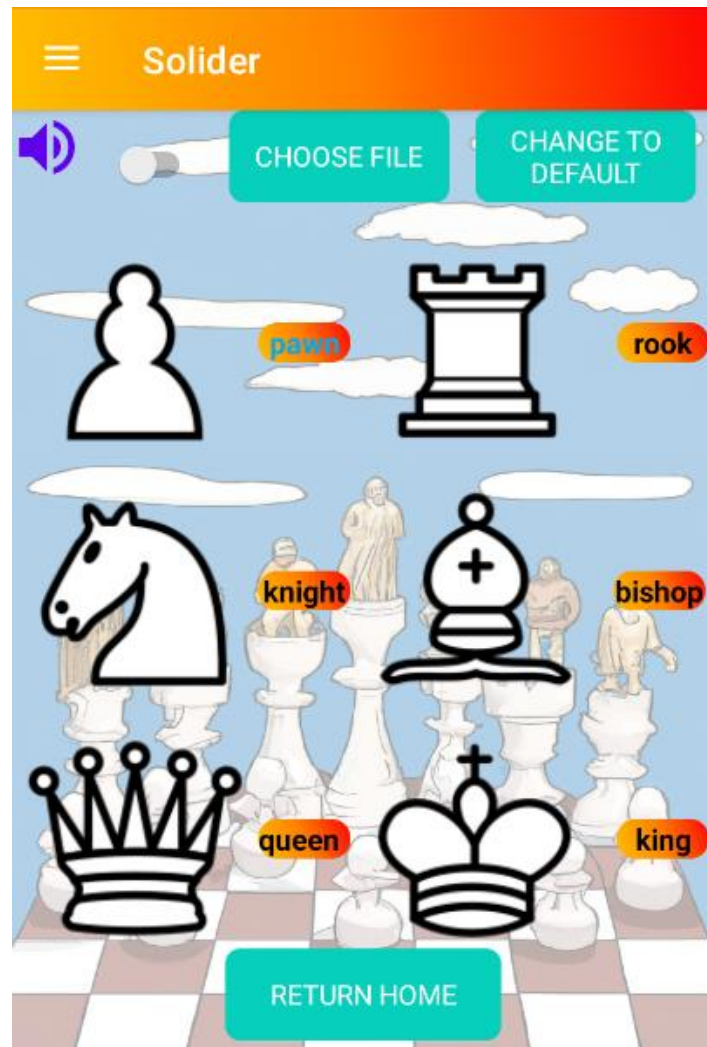


מסך Soldier

מסך זה הינו Fragment

- כפתור Return home יעביר את האפליקציה למסך הבית ויעדכן אותו לצבעי הלוח החדשים.
- כפתור לעצירת והמשכת המוזיקה.
- כפתור Choose file ייתן למשתמש תמונה מהגלריה שלו או מהמצלמה, המשתמש יבחר בעזרת דיאלוג שייפתח. תמונה זו תחליף את החייל ששמו מסומן בכחול לתמונה החדשה. התמונה תישמר בפייר סטור ולבסוף יוקפץ טוסט שיגיד שהתמונה הועלתה.
- בעת לחיצה על אחד החיילים שמו בלבד יסומן בכחול וברגע בחירת התמונה תמונתו תוחלף.
- המתג ישנה את צבעי החיילים משחור ללבן וההפך.
- הכפתור change to default ישנה את תמונת החייל המסומן לתמונה המקורית של אותו חייל.

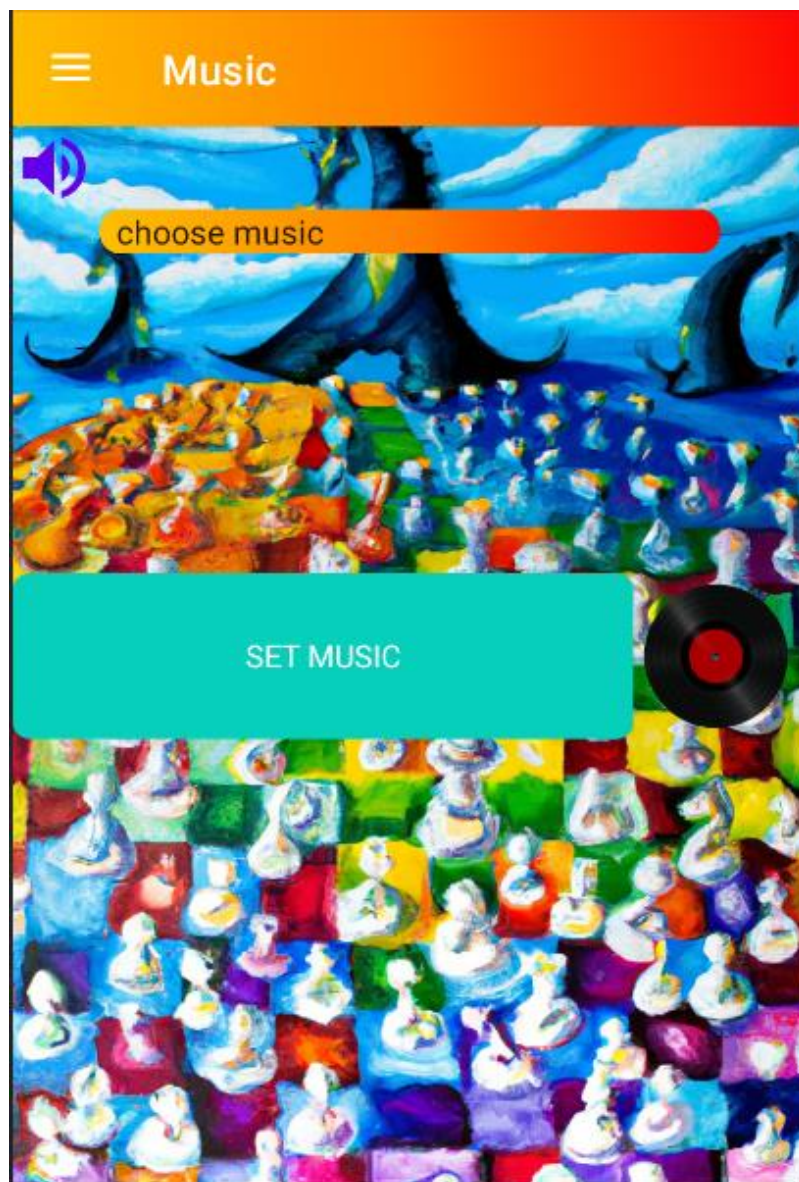




מסך Music

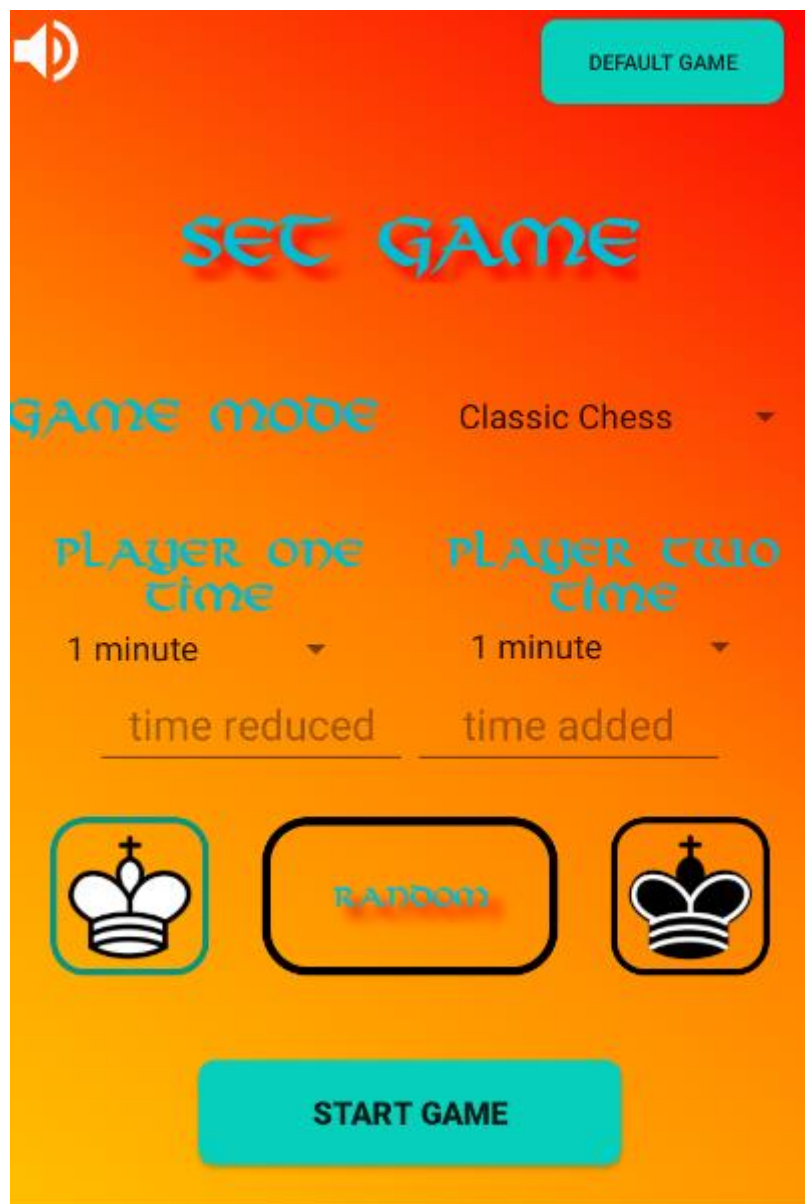
מסך זה הינו Fragment

- כפתור לעצירת והמשכת המוזיקה.
- Spinner בו ניתן לבחור שיר מסוים מתוך אוסף קיים.
- הכפתור בצורת התקליט ישמיע את המוזיקה שנבחרה ללא הגדרתה כמוזיקת הרקע. בעת לחיצה לבדיקת המוזיקה הוא יסתובב.
- כפתור set music יגדיר את המוזיקה שנבחרה למוזיקת הרקע.



מסך Set game

- כפתור לעצירת והמשכת המוזיקה.
- Spinner בו ניתן לבחור את סוג המשחק.
- Spinner בהם ניתן לבחור את כמות הזמן של כל שחקן.
- שלושה כפתורים המגדירים איזה מהצבעים יתחיל: לבן, ירוק או בחירה אקראית.
- כפתור default game בו אוטומטית יאותחלו ערכים ספציפיים.
- כפתור start game שפותח את מסך לוח השחמט במידה והמשתמש בחר בשחמט רגיל, ובעת בחירה של שחמט מעוצב יפתח את חלון סידור הלוח.



מסך Set board

- שני EditText אשר בהם ייכתב גודל הלוח החדש רוחבXאורך.
- כפתור apply בוא יעודכן לוח הדוגמה לגודל הלוח העדכני (חיילים בלוח הישן ימחקו).
- לוח שנמצע ב scroll view אשר גודלה שווה לגודל שהשתמש בחר. על הלוח ישובצו החיילים שנבחרו. הלוח הוא הקדמה לאיך יראה הלוח בתחילת המשחק.
- שני שורות של חיילים שאותם יוכל השחקן לבחור (החייל המסומן ייצבע בכחול) ולבחור נקודה בלוח שם יניח את החייל הספציפי.
- כפתור מחיקה. כאשר לוחצים עליו ואז על החיילים, הם נמחקים. יציאה ממצב מחק בעזרת לחיצה נוספת על הכפתור.
- כפתור default בו אוטומטית יאותחל הלוח ל8X8 עם המיקום ההתחלתי של כל החיילים במשחק שחמט קלאסי.

שפותח את מסך בחירת והאכילה. האפליקציה שישאל את המשתמש שהוא רוצה לעבור מסך. המסך כאשר יש פחות צבע וכאשר המלכים צמודים אחד לשני.



- כפתור save התנועה תציג דיאלוג אם הוא בטוח לא יתחלף ממלך אחד לכל השונים



מסך Set movement

- שורה עם החיילים שנבחרו במסך הקודם. בעת לחיצה החייל שנלחץ בלבד ישנה את צבעו לכחול ויופיע במרכז הלוח.
- לוח בגודל הנבחר במסך הקודם (scroll View). כאשר נבחר חייל הוא יופיע במרכז הלוח. בעת לחיצה על משבצת מסוימת היא תהפוך למסומנת.
- כפתור Add מוסיף את הנקודה למערך התנועות/ ההתקפות של החייל הספציפי ומציג אותו למעלה.
- Switch בוחר אם הנקודה החדשה היא התקפה או תנועה.
- כפתור Remove מסיר את המטריצה האחרונה מהסוג שנבחר במתג.
- כפתור Default יעדכן את התנועה וההתקפה לערך ברירת המחדל בהתאם לסוג החייל.
- Radio button יגדירו אם התנועה/ההתקפה של החייל סופית ו/או אם הוא יכול לקפוץ מעל חיילים אחרים.

Finish ישמור את התנועות/ של החייל ויסיר את הסימון משורת החיילים. Continuen יעביר את האפליקציה למסך לוח ויתחיל את המשחק.

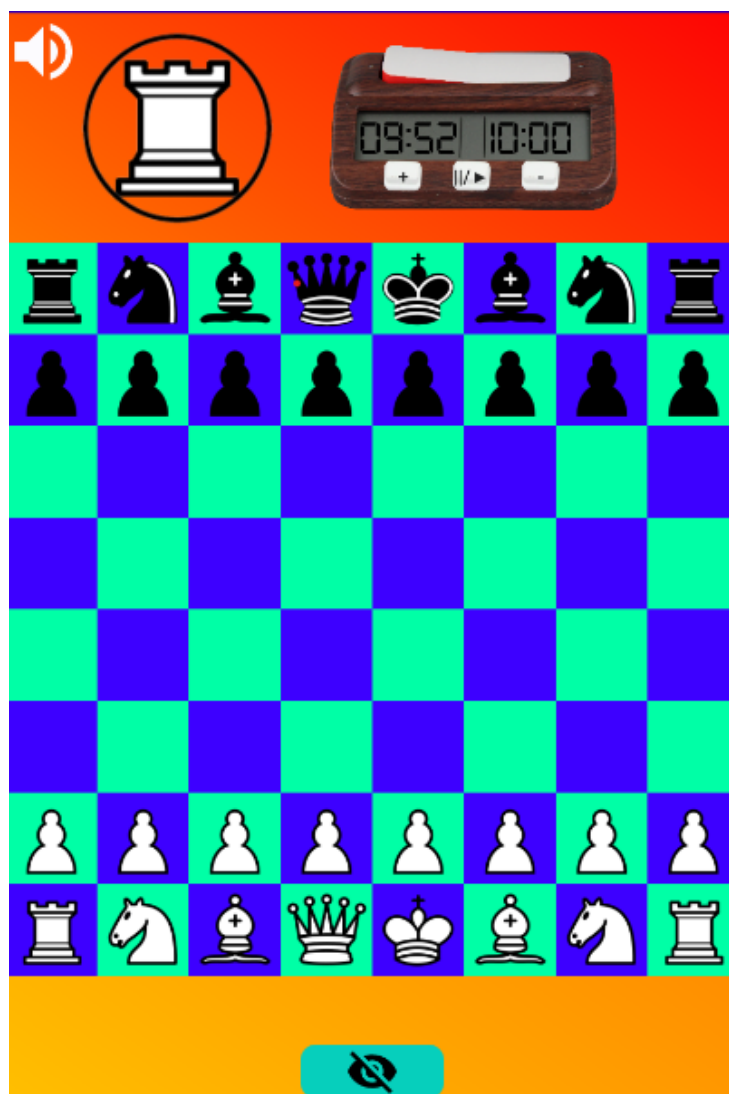


- כפתור ה התקפות שלו
- כפתור המשחק



gameboard מסך

- כפתור לעצירת והמשכת המוזיקה.
- טיימר שמתחיל מכמות הדקות שהוגדרה במסך הקודם ומעדכן את הזמן של השחקן שתורו כל שנייה.
- בטיימר אופציה לעצור את מדידת הזמן (הלוח יקפא ושום תנועה לא תתאפשר), להוסיף את זמן בכמות שהוגדרה מראש ולהוריד זמן בכמות שהוגדרה מראש. כאשר יגיע השעון ל00:00 המשחק ייגמר.
- סמל של צריח המראה את תור השחקן הנוכחי. כל תור ישנה את צבעו לצבע השחקן שתורו.
- לוח עם חיילים כפי שסודרו מראש. מאפשר את הפונקציות של המשחק.
- כפתור בצורת עין שבעת לחיצה מראה את התמונות המקוריות של החיילים. כאשר הכפתור ילחץ שוב יחזרו החיילים לתמונה שהוגדרה להם מהגלריה של המשתמש.

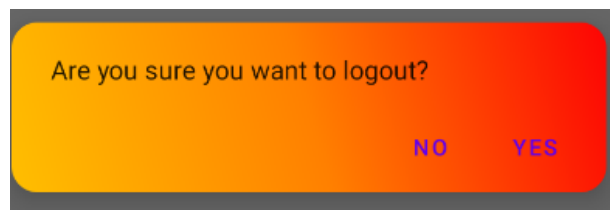


הודעות למשתמש

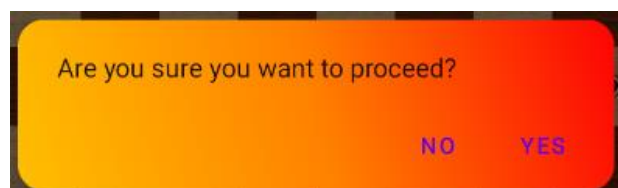
יש לפרט כאן על ההודעות שהאפליקציה מציגה למשתמש – Notifications ,Toasts ,Dialogs .
יש להוסיף צילום מסך עבור כל הודעה.

בחלון הבית :homeActivity

בדיקה אם המשתמש בטוח שהוא רוצה להתנתק

**בחלון בחירת גודל הלוח :CustomSetBoard**

בדיקה אם המשתמש בטוח שהוא רוצה להמשיך לחלון הבא



וגם טוסטים:

בדיקה אם המשתמש עמד בתנאים לבניית לוח משחק עם שחקנים.

"pawns cannot stand on the first or last rows"

"must select at least one king from each color"

"different kings are one tile from each other"

"board size is to small"

בחלון בחירת ערכי המשחק :SetGame

בדיקה אם הערכים שהזין המשתמש תקינים (טוסט)



```
"enter value to add/reduce time"
```

```
"add/reduce value should be positive"
```



בחלון לוח המשחק gameBoardActivity

בעת בדיקת שח-מט ותיקו יוקפצו טוסטים:

"Can attack"

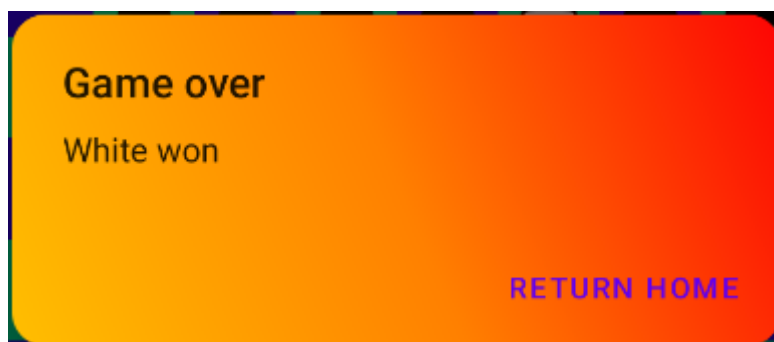
"attacker can be attacked"

"Can escape"

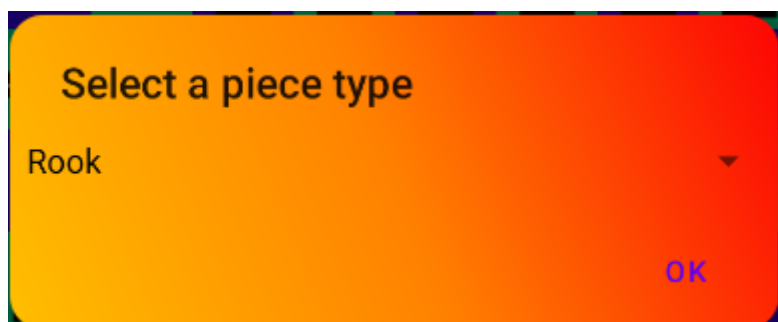
"checkmate"

"draw"

בעת סיום משחק יוקפץ דיאלוג בוא יוכרז המנצח או תיקו:



כאשר יצעד פיון לסוף הלוח יוקפץ דיאלוג בו יבחר המשתמש את סוג החייל אליו יהפוך הפיון:



בחלון בחירת התנועה/ תקיפה :setMovementActivity

בדיקה אם יש את אותה התנועה/התקפה פעמיים, אם הועלתה ההתקפה/תנועה, ואם המשתמש ניסה לעבור לחייל אחר לפני ששמר את התנועות/התקפות של החייל הנוכחי. (טוסטים)

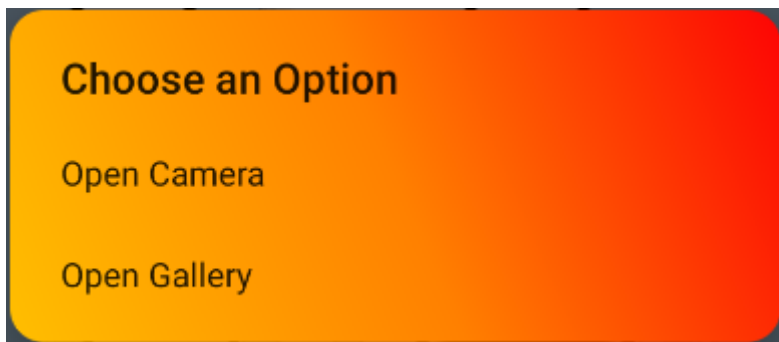
"movement already picked"

"attack already picked"

"uploaded!"

"finish setting current soldier"

תמונה לחיילים



בחלון בחירת

:soldierFragment

דיאלוג הנותן למשתמש לבחור אם ירצה להעלות תמונה מהגלריה או מהמצלמה שלו.

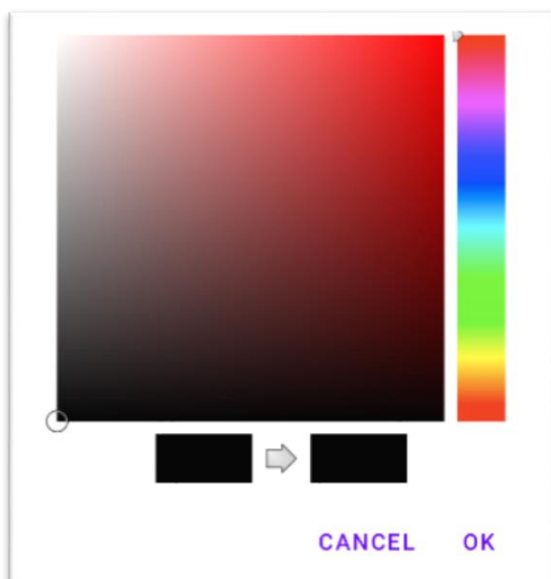
התמונה הועלתה בהצלחה מהגלריה. (טוסט)

Upload successful





בחלון :boardFragment
דיאלוג בחירת צבע



תיאור תחום הידע

בחלק זה נפרט את האובייקטים הנחוצים לאפליקציה, אותם אובייקטים ומחלקות שכתבנו ביחד כשישבנו לתכנן את הפרויקט.

שם האובייקט:

עבור משחקים: רשימת דמויות, כלים, לוחות משחק, שחקנים, קופה וכו'.
עבור אפליקציות מנהליות: תיאור סוגי משתמשים, מידע בשימוש, סוגי טבלאות וכו'.

סוג הנתונים:

מה האובייקט מייצג, מה הוא שומר.

ייצוג המידע:

עבור משחקים: ייצוג לוח המשחק, תכונותיו, ייצוג מצב בלוח.
עבור אפליקציות מנהליות: ייצוג מבני הנתונים, תכונותיהם, קשרים בין הנתונים (אם קיימים) ונימוק.
דוגמאות: רשימה, מערך דו ממדי, עץ בינארי, טבלאות וכו'. מדוע נעשה שימוש בייצוג זה.

תיאור פעולות על המידע:

עדכון מצב, קריאת מצב, מחיקה והוספה.



שם האובייקט: Soldier

סוג הנתונים: מחלקה אבסטרקטית שממנה יורשים כל סוגי החיילים במשחק. שומר את המידע הבסיסי של כל חייל.

ייצוג המידע:

```
protected Bitmap imageBitmap;
protected boolean whitePiece;
protected Bitmap defaultImageBitmap;
protected int[] startPosition = new int[2]; // [row,col]
protected int[] currentPosition = new int[2]; // [row,col]
protected int[][] validMovement;// [[row,col],[row,col]]
protected int[][] validAttack;// [[row,col],[row,col]]
protected boolean infiniteMovement;
protected String name;
protected boolean canRunOver;
protected boolean canAttackOver;
protected boolean infiniteAttack;
protected static int rows;
protected static int cols;
protected static boolean customChess = false;
```

- שמירת התמונה ממסד הנתונים בענן של כל חייל.
- שמירת צבע החייל.
- שמירת תמונת ברירת המחדל של כל חייל.
- המיקום ההתחלתי של כל חייל.
- המיקום הנוכחי של כל חייל.
- מטריצת התנועות האפשריות של כל חייל.
- מטריצת ההתקפות האפשרות של כל חייל.
- האם התנועה של החייל אינסופית?
- שם החייל.
- האם החייל יכול לדלג מעל חיילים אחרים?
- האם החייל יכול לתקוף מעל חיילים אחרים?



- האם ההתקפה של החייל אינסופית?
- השורה הנוכחית של החייל.
- העמודה הנוכחית של החייל.
- האם סוג המשחק הוא קלאסי או custom? (משתנה סטטי)



תיאור פעולות על המידע: למחלקה פעולות אבסטרקטיות :

```
public abstract boolean IsValidMovement(BoardTile currentTile, BoardTile[][] boardTileMatrix);  
public abstract boolean IsValidAttack(BoardTile currentTile, BoardTile[][] boardTileMatrix);  
public abstract void highlight(BoardTile[][] boardTileMatrix,boolean highlight);
```

אשר נועדו לבדוק אם תנועה של החייל למשבצת מסוימת אפשרית, אם תקיפה של חייל במשבצת מסוימת אפשרית ואיפה כל המשבצות אליהם יכול להגיע החייל.



שם האובייקט: BoardTile

סוג הנתונים: BoardTile יורש מ `androidx.appcompat.widget.AppCompatImageView`

המחלקה מייצגת משבצת, גם מבחינה ויזואלית.

```
public int row;
public int col;
Soldier currentSoldier = null;
int length;
private boolean redAura = false;
int tileColor;//-2 for default brown -1 for default white
boolean highlight = false;
```

השורה והעמודה עליה היא נמצאת.

החייל שנמצא עליה.

האורך של משבצת (חישוב של כמות המשבצות חלקי אורך המסך).

האם המשבצת צריכה להיות אדומה.

צבע המשבצת.

האם המשבצת צריכה להיות מסומנת.

ייצוג המידע: לוח המשחק מורכב ממערך של BoardTile ועליו מתבצעות התנועות בלוח.

תיאור פעולות על המידע: פעולת `onDraw` שמציירת מחדש את המשבצת; בודקת אם לצייר רקע כחול או אדום וחייל.



שם האובייקט: Board

סוג הנתונים: יוצר ומייצג את הלוח (מטריצת BoardTile). Board יורש מLinearLayout

ייצוג המידע:

```
int cols;  
int rows;  
BoardTile[][] boardTileMatrix;  
ArrayList<Soldier> soldiers;
```

כמות השורות והעמודות של הלוח.

הלוח עצמו.

כל החיילים על הלוח.

תיאור פעולות על המידע: מחשב את האורך הדרוש כדי שrows שורות יכנסו באורך של המכשיר ויוצר את

הלוח.



שם האובייקט: `GameActivity`

סוג הנתונים: מנהל את המשחק; מזיז שחקנים, בודק שח ומט, מחליף את התור, מסובב את הלוח, דואג לזמנים וכו....

ייצוג המידע:

```
private boolean isWhiteTurn = true; //white always starts
int[] blackTime = new int[2];
int[] whiteTime = new int[2];
TextView playerOneTime;
TextView playerTwoTime;
boolean pauseGame = false;
Context context;
ImageView imageView;
boolean whiteKingChecked = false;
boolean blackKingChecked = false;
ImageView showNames;
LinearLayout boardLayout;
BoardTile[][] boardTileMatrix;
ArrayList<Soldier> soldiers;
ArrayList<Soldier> constSoldiers = new ArrayList<>();
BoardTile selectedBoardTile;
Soldier whiteKing, blackKing;
Soldier checkMateorToBlackKing, checkMateorToWhiteKing; // the piece which threaten the king
float rotationCounter = 0;
int turnsCounter = 1;
private Toast toast;
int changelImageCounter = 0;
```

בודק מי מתחיל.

כמות הזמן של הצד השחור והלבן.

עדכון הזמן של הטימר.

האם לעצור את המשחק.

הcontext של מסך לוח המשחק.



התמונה של הצריח שמעיד על צבע השחקן הנוכחי.

האם המלך הלבן והשחור מאוימים.

הראה את התמונות המקוריות.

הLinearLayout שבה יוצג הלוח.

הלוח עצמו.

כל החיילים על הלוח.

כל החיילים שהיו בהתחלת המשחק.

מצביע על שני המלכים.

מצביע על מי שמאיים על המלכים.

Counter לבדיקת כמות המעלות שצריך לסובב את הלוח.

מספר התורות שבוצעו

טוסט

כמות הפעמים שכפתור שינוי התמונה (בצורת העין) נלחץ

תיאור פעולות על המידע: תפעול המשחק – תנועה, שח, מט, טיימר וכו'...



שם האובייקט: User

סוג הנתונים:

```
private String username;  
private String password;  
private UserCustomize userCustomize;
```

שם משתמש.

סיסמה.

מופע של UserCustomize.

ייצוג המידע: תנאים מזהים.

תיאור פעולות על המידע: קריאת שם המשתמש, סיסמה והתמונות של החיילים.

שם האובייקט: UserCustomize

סוג הנתונים:

```
public int playerOneColor = R.drawable.brighttile;  
public int playerTwoColor = R.drawable.browntile;  
  
public int currentAudio = R.raw.dragonforce;  
  
public int wpawn = R.drawable.wpawn;  
public int wrook = R.drawable.wrook;  
public int wknight = R.drawable.wknight;  
public int wbishop = R.drawable.wbishop;  
public int wqueen = R.drawable.wqueen;  
public int wking = R.drawable.wking;
```



```
public int bpawn = R.drawable.bpawn;  
public int brook = R.drawable.brook;  
public int bknight = R.drawable.bknight;  
public int bbishop = R.drawable.bbishop;  
public int bqueen = R.drawable.bqueen;  
public int bking = R.drawable.bking;
```

צבעים של הלוח.

השמע הנוכחי.

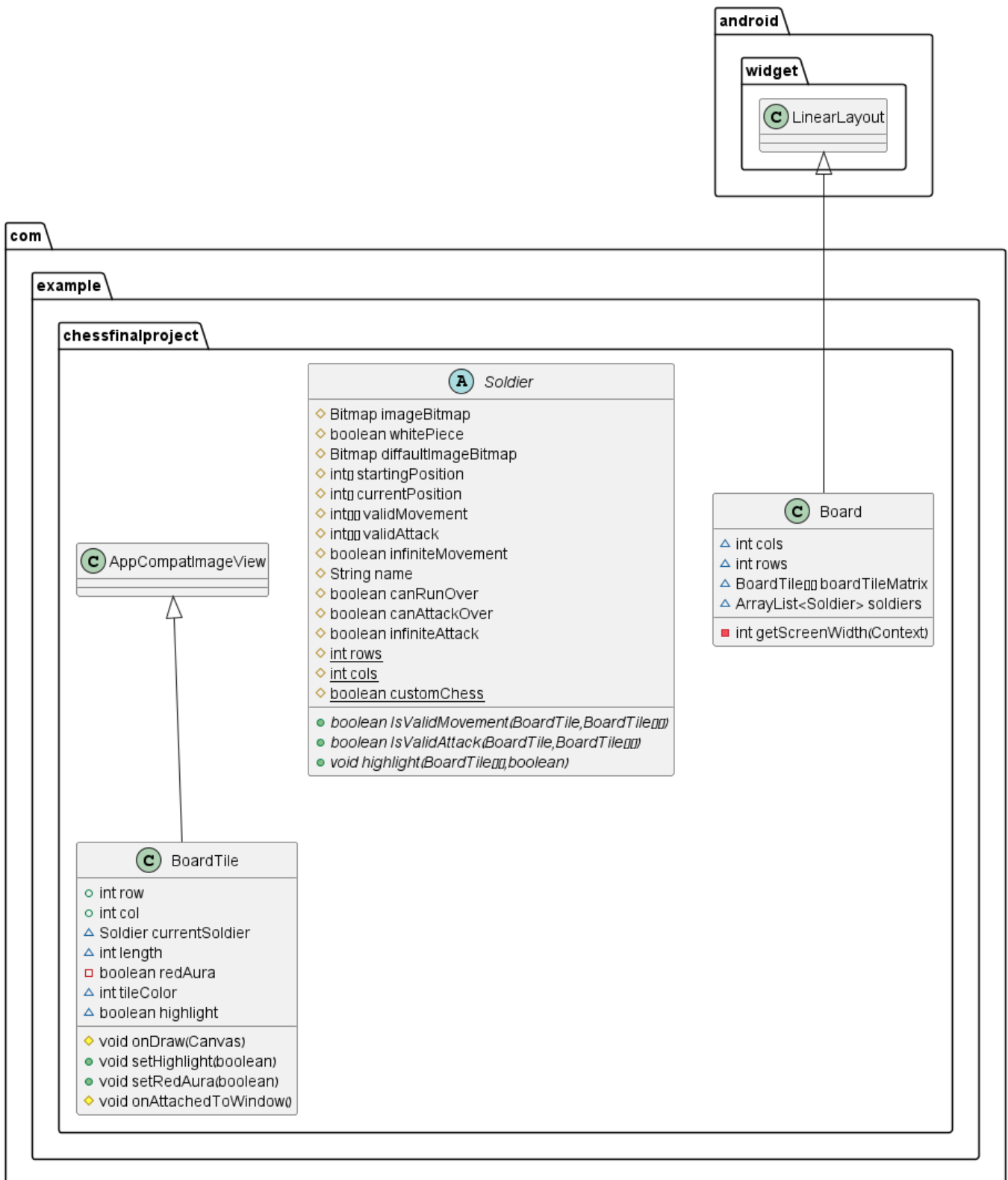
התמונות של כל החיילים

ייצוג המידע: תמונות אישיות של המשתמש.

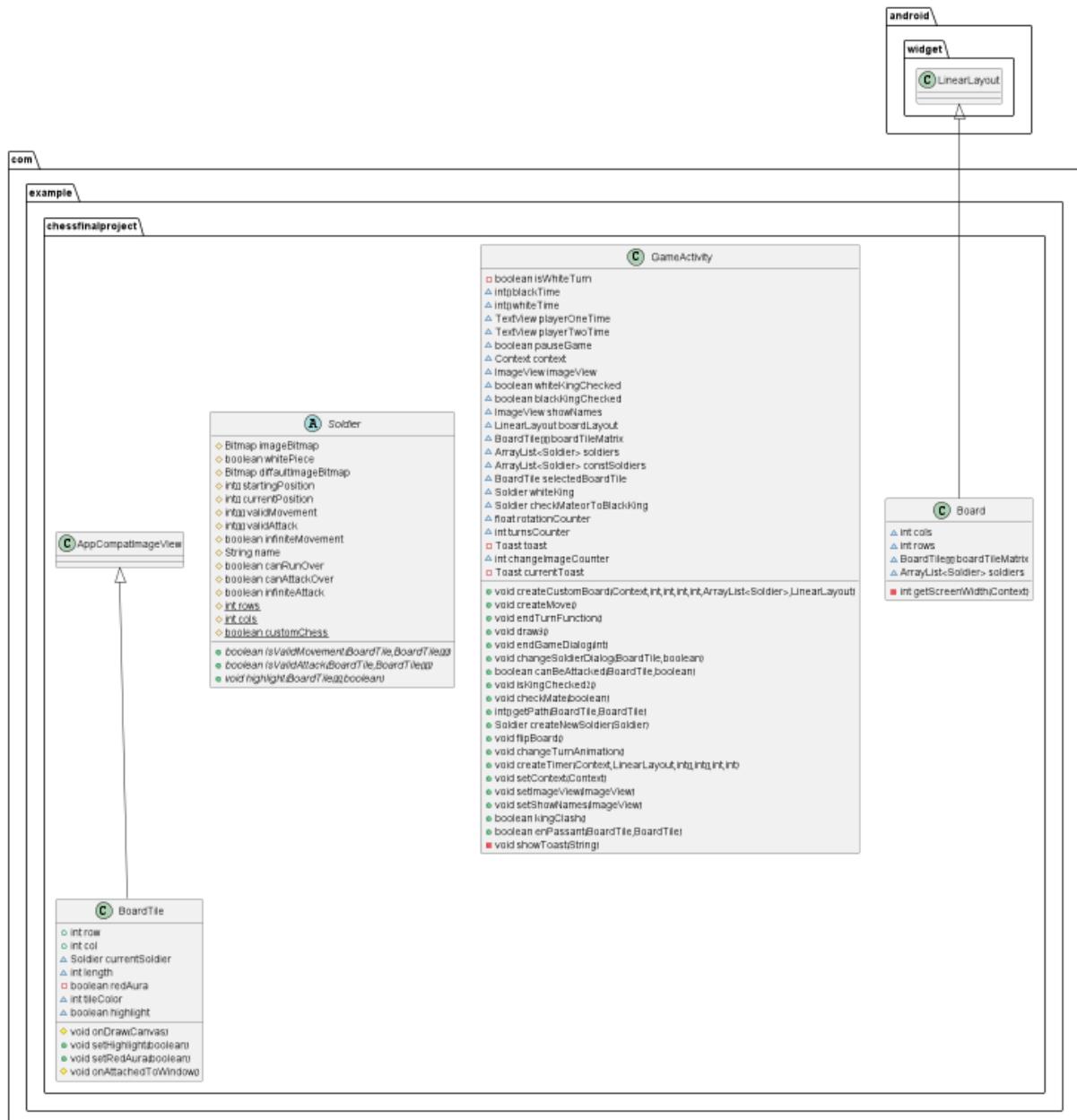
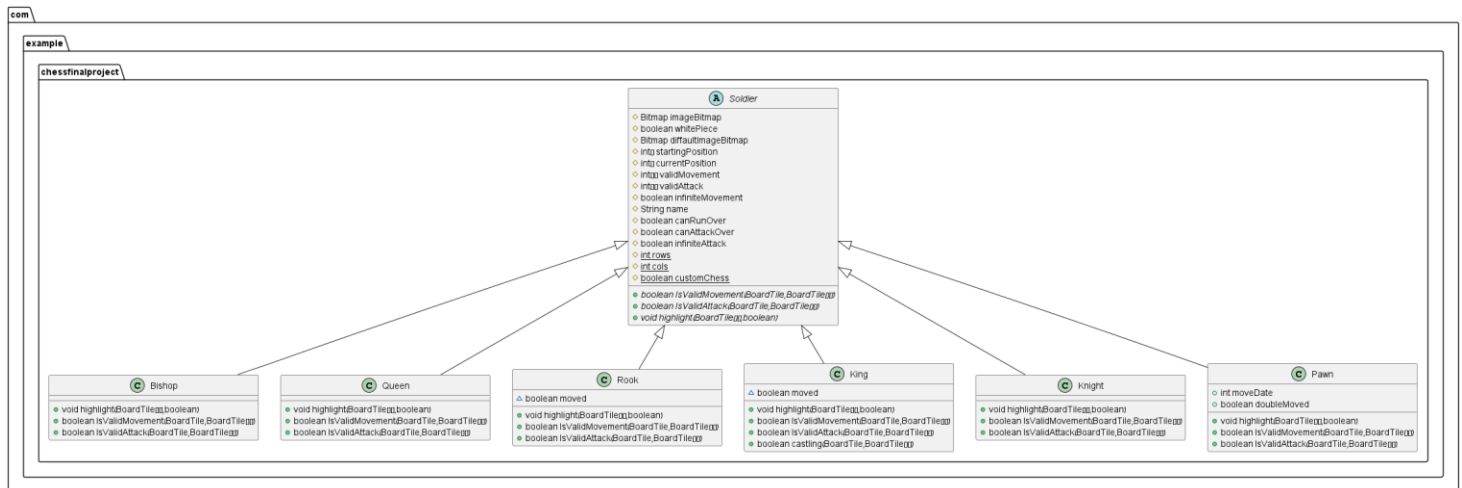
תיאור פעולות על המידע: קריאת הצבעים והתמונות והשמטת כלוח השחמט כחיילים ומשבצות.



תרשים מחלקות (UML)







מימוש הפרויקט

מטרת הפרק הזה היא להסביר למפתח את דרך מימוש הפרויקט, ממש כמו תיקי הפרויקט שקיבלתם ב- Go Fish!, Elements Fight! ו-FallBall. הפרק מורכב משני חלקים עיקריים: תיאור המחלקות ותיאור אחסון הנתונים.

מחלקות הפרויקט

עבור כל המחלקה יש לציין את הדברים הבאים:

- שם המחלקה
- תפקיד המחלקה
- תיאור תכונות עיקריות
- תיאור פעולות עיקריות

הדגש בתיאור יש לעשות על מחלקות המודל – מחלקות שפתחתם בעצמכם שלא קשורות לשום אלמנט ויזואלי: דמויות, לוגיקת האפליקציה, מחלקות שמממשות אלגוריתמים, מחלקות שרות מקוריות (Services) וכו'.

חשוב לציין שאין צורך לתאר כל התכונות וכל הפעולות של המחלקה. אין צורך לתאר פעולות get/set טריוויאליות, משתני עזר, פעולות מערכת כמו onCreate, אם אין עליהם תפקיד חשוב מעבר ליצירת אובייקטים וקישור ל-XML, אין צורך לתאר מאזינים פשוטים וכו', אם במחלקה מספר פעולות שמטפלות באובייקטים דומים מרובים, אפשר לתאר את כל הפעולות ביחד.

מומלץ לתאר כל מחלקה בעמוד נפרד.

הדוגמאות בעמוד הבא.



הסבר על מימוש התנועה והמתקפה

הרעיון המרכזי שעליו בוסס מערכות התנועה והאכילה הם מטריצות של שורה ועמודה: לכל חייל יש מטריצה של מטריצות שבהם שמורות כל סוגי התנועות והמתקפות (במטריצה אחרת).

לדוגמא, חייל עומד בשורה 4 ועמודה 5 (השורה העליונה ביותר היא שורה 0, העמודה השמאלית ביותר היא עמודה 0) – המיקום הנוכחי שלו יישמר במטריצה בגודל 2×2 כ-[4,5]. נניח שמטריצת התנועות שלו שווה ל- $\begin{bmatrix} 1,1 \\ 1,0 \end{bmatrix}$, $\begin{bmatrix} 1,-1 \end{bmatrix}$, בעת בחירת משבצת חדשה אליה יזוז החייל האפליקציה תבדוק אם התנועה אפשרית על ידי הוספת ערכי השורה-עמודה למיקום הנוכחי של החייל ובדיקה אם המשבצת החדשה עומדת בחוקיות. לדוגמא נרצה שהחייל ב-[4,5] יזוז ל-[5,4]:

- נבדוק את התנועה הראשונה $[1,1]$: $5=4+1$, $6=5+1$. לא שווה ל-[5,4] ולכן החייל לא יכול ללכת לנקודה החדשה בעזרת החוקיות הזאת, נעבור לתנועה הבאה.
- נבדוק את התנועה הראשונה $[0,1]$: $6=5+1$, $4=4+0$. לא שווה ל-[5,4] ולכן החייל לא יכול ללכת לנקודה החדשה בעזרת החוקיות הזאת, נעבור לתנועה הבאה.
- נבדוק את התנועה הראשונה $[1,-1]$: $4=5-1$, $5=4+1$. שווה ל-[5,4] ולכן החייל יכול ללכת לנקודה (- מבחינה חוקית, האפליקציה צריכה לבדוק אם המהלך תקין מבחינת חוקי המשחק).

כמו כן, המשתמש יכול לבחור אם התנועות או ההתקפות יוכלו להתבצע מעבר לחיילים (דילוג מעל חייל, כמו פרש) והאם התנועה שלהם סופית או אין סופית (מלך לעומת מלכה).



מחלקת Soldier

המחלקה היא אבסטרקטית וממנה יורשים כל מחלקות השחקנים.

תכונות (מאפיינים)	
protected Bitmap imageBitmap	מחזיק את התמונה של החייל.
protected int[] currentPosition	מחזיק את המיקום הנוכחי של החייל על הלוח: [עמודה, שורה].
protected boolean whitePiece;	מחזיק את צבע החייל
protected int[][] validMovement;	מחזיק את כל סוגי התנועות שיכול לבצע החייל: [מס' עמודות ימין, מס' עמודות למטה].
protected int[][] validAttack;	מחזיק את כל סוגי ההתקפות שיכול לבצע החייל: [מס' עמודות ימין, מס' עמודות למטה].
פועלות	
public abstract boolean IsValidMovement(BoardTile currentTile, BoardTile[][] boardTileMatrix);	בודק אם התנועה של החייל למשבצת מסוימת חוקית מבחינת התנועות האפשריות שלו.
public abstract boolean IsValidAttack(BoardTile currentTile, BoardTile[][] boardTileMatrix);	בודק אם התקיפה של החייל למשבצת מסוימת חוקית מבחינת ההתקפות האפשריות שלו.
public abstract void highlight(BoardTile[][] boardTileMatrix, boolean highlight);	סמן עיגול כחול על כל משבצת שאליה החייל יכול לנוע/לתקוף, או מחק את העיגול הכחול.





מחלקת BoardTile

המחלקה יורשת מ `androidx.appcompat.widget.AppCompatImageView`

כל מופע של המחלקה הוא משבצת בלוח השחמט, עליו נעשות התנועות במשחק.

תכונות (מאפיינים)
<code>public int row;</code> מחזיק את השורה של המשבצת הנוכחית.
<code>public int col;</code> מחזיק העמודה של המשבצת הנוכחית.
<code>Soldier currentSoldier;</code> מחזיק את המופע של החייל העמוד על המשבצת.
<code>int tileColor;</code> מחזיק את צבע המשבצת.
<code>protected int[][] validAttack;</code> מחזיק את כל סוגי ההתקפות שיכול לבצע החייל: [מס' עמודות ימינה, מס' עמודות למטה].
פועלות
<code>@Override</code> <code>protected void onDraw(Canvas canvas)</code> פעולה המציירת את המשבצת על הלוח.



מחלקת Board

המחלקה יורשת מ LinearLayout

המופץ של המחלקה הוא הלוח.

תכונות (מאפיינים)
<pre>public int rows;</pre> <p>מחזיק את מספר השורות של הלוח.</p>
<pre>public int cols;</pre> <p>מחזיק את מספר העמודות של הלוח.</p>
<pre>BoardTile[][] boardTileMatrix;</pre> <p>מחזיק את "הלוח". מטריצה של משבצות.</p>
<pre>ArrayList<Soldier> soldiers;</pre> <p>מחזיק את כל החיילים שבלוח בעת יצירתו.</p>
פועלות
<pre>public Board(Context context, int cols, int rows,int colorOne, int colorTwo, ArrayList<Soldier> soldiers)</pre> <p>הבנאי של המחלקה. הבנאי מחשב את אורך המשבצת בהתאם למספר השורות ורוחב המכשיר. לבסוף הוא בונה את הלוח.</p>



מחלקת GameActivity

המחלקה מנהלת את המשחק עצמו.

תכונות (מאפיינים)
<pre>ArrayList<Soldier> soldiers;</pre> <p>מחזיק את המצביעים של כל החיילים על הלוח.</p>
<pre>Soldier whiteKing, blackKing;</pre> <p>מחזיק את המצביעים של החיילים מסוג מלך.</p>
<pre>BoardTile[][] boardTileMatrix;</pre> <p>מחזיק את "הלוח". מטריצה של משבצות.</p>
<pre>ArrayList<Soldier> soldiers;</pre> <p>מחזיק את כל החיילים שבלוח בעת יצירתו.</p>
<pre>int[] blackTime = new int[2];</pre> <pre>int[] whiteTime = new int[2];</pre> <p>מחזיק את הזמן שנותר לכל שחקן: [שניות, דקות]</p>
<pre>boolean whiteKingChecked = false;</pre> <pre>boolean blackKingChecked = false;</pre> <p>מחזיק בכל רגע האם יש שח על כל מלך.</p>
פועלות
<pre>public void createCustomBoard(Context context, int cols, int rows, int colorOne, int colorTwo, ArrayList<Soldier> soldiers, LinearLayout boardLayout)</pre> <p>יוצר מופע של Board ובונה את לוח המשחק.</p>



```
public void createMove()
```

יוצר onClickListener לכל משבצת על הלוח. בודק אם כל תנועה שהשחקן מבצע חוקית או לא ובמידת הצורך מבטל אותה. בודק בכל צעד אם המשחק נגמר או לא.

```
public void isKingChecked2()
```

בודק אם יש שח ואם יש שחמט.

```
public void flipBoard()
```

הופך כל משבצת ב-180 מעלות.

```
public void changeTurnAnimation()
```

מייצר את האנימציה של סיום התור.

```
public void createTimer(Context context,LinearLayout layout,int[] blackTime ,int[]
whiteTime,int addTime,int reduceTime)
```

יוצר את התמונה של שעון העצר ומעדכן אותו.

```
public void setShowNames(Imageview showNames)
```

יוצר את הכפתור שמחליף את התמונה מתמונת ברירת המחדל לתמונה שבחר המשתמש.

בסיס נתונים מרוחק (Firebase)

- בתחילה יש לציין איזה סוג בסיס נתונים: authentication, Realtime database או Firestore
- עבור כל בסיס נתונים יש להסביר כל ענף (child) בנפרד.
- עבור כל ענף יש להגדיר את הדברים הבאים:
 1. מפתח הענף
 2. תפקיד הנתונים הנמצאים בענף
 3. הסבר למבנה הנתונים בענף- הסבר לכל מפתח
 4. להוסיף תמונה של ענף מתוך Firebase Console
 5. אם נעשה רישום המשתמשים יש לציין את שיטת הרישום

בסיס נתונים מסוג Realtime database



מפתח	סוג נתון	תיאור הנתון
username	String	שם משתמש
password	String	סיסמה מוצפנת
userCustomize	UserCustomize	צבע המשבצות (int)

שיטת שמירת משתמשים

לכל משתמש סיסמה מוצפנת בעזרת Md5. יכול להיות רק משתמש אחד בעל שם משתמש ספציפי.

```
https://chess-champion-a7ff6-default-rtdb.firebaseio.com/  
└─ users  
  └─ tomer25  
  └─ tomer27  
    └─ password: "e10adc3949ba59abbe56e057f20f883e"  
    └─ userCustomize  
      └─ playerOneColor: -2370091  
      └─ playerTwoColor: -13816531  
    └─ username: "tomer27"
```



בסיס נתונים מסוג Firestore

בסיס הנתונים מכיל שני סוגים של נתונים: תמונות ברירות מחדל של חיילים ותמונות השחקנים האישיות של כל משתמש.















תמונות ברירות המחדל שמורות כך:

Color + piece type.png

לדוגמא, פיון לבן יהיה שמור כך:

wpawn.png

תמונות שחקן אישיות יהיו שמורות כך:

gs://chess-champion-a7ff6.appspot.com				
<input type="checkbox"/>	Name	Size	Type	Last modified ↑
<input type="checkbox"/>	 bbishop.png	1.92 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 bking.png	3.32 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 bknight.png	2.39 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 brook.png	1.08 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wrook.png	1.26 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wbishop.png	2.73 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wpawn.png	1.96 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wqueen.png	4.28 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wknight.png	2.85 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 bqueen.png	3.53 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 bpawn.png	1.31 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 wking.png	3.19 KB	image/png	Mar 19, 2023
<input type="checkbox"/>	 tomer2brook.jpg	134.3 KB	image/jpeg	Mar 19, 2023
<input type="checkbox"/>	 tomer2wbishop.jpg	55.91 KB	image/jpeg	Mar 21, 2023



Username + color + piece type.png

לדוגמא, צריח שחור של המשתמש tomer2:

tomer2brook.png



SharedPreferences

יש לציין את הפרטים הבאים:

1. שם הקובץ
2. איפה נשמר – זיכרון פנימי/חיצוני (תמיד פנימי)
3. עבור כל ערך יש לציין את שמו, סוג המידע ותפקידו

שם הקובץ: UsernamePref

מיקום: זיכרון פנימי

מפתח	סוג	תיאור הנתון
username	String	שומר את שם המשתמש בעת כניסה ראשונית עד התנתקות. כאשר הערך אינו ריק, המשתמש יוכנס אוטומטית בלי להכניס סיסמה. כאשר המשתמש מתנתק ישונה הערך של המשתנה לnull.



Services

עבור כל Service שאתם יצרתם באפליקציה שלכם כתבו:

- שם ה-Service
- סוג ה-Service (Background / Foreground)
- מה תפקידו, מה הוא עושה
- מתי מופעל ומתי נעצר

שם: MusicService

סוג: Background Service

שימוש: ה-Service משמש להשמעת מוזיקה ברקע האפליקציה ללא תלות במסך הנוכחי.

מתי מופעל ומתי נעצר: ה-Service מופעל כאשר המשתמש מועבר למסך הבית לאחר התחברות. הservice לא

מושמד עד סגירת האפליקציה.



Camera & Gallery

שימוש: העלאת/צילום תמונה והכנסתה ל Fire storage כתמונה שתיוצג חייל מסוים בסט החיילים של המשתמש הנוכחי.

שם הקובץ בו נמצא: SoldierFragment.java

העלאת התמונה מהגלריה

העלאת התמונה מהגלריה מתבצעת בעזרת `ActivityResultLauncher<String>`:

```
private ActivityResultLauncher<String> pickImageLauncher;
```

כאשר רוצה המשתמש לבחור תמונה מהגלריה נפעיל את `pickImageLauncher` ונסנן את הקבצים כך שיוצא למשתמש לבחור רק קבצי תמונות:

```
pickImageLauncher.launch("image/*");
```

לאחר פקודה זו תופעל הפעולה לה עשינו `register`.

צילום התמונה בעזרת מצלמה

פתיחת המצלמה של המכשיר אפשרי בעזרת `ActivityResultLauncher<Intent>`

```
ActivityResultLauncher<Intent> activity01Launcher;
```

כאשר בחר המשתמש תמונה שצילם נפעיל את `activity01Launcher`:

```
Intent cameraIntent = new  
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);  
activity01Launcher.launch(cameraIntent);
```



לאחר פקודה זו תופעל הפעולה לה עשינו register.



רכיבים מתקדמים נוספים

כתבתי לכם כאן את רשימת הנושאים המתקדמים מתוך המחווה של משרד החינוך. עבור כל רכיב מתקדם שנמצא ברשימה ודורש פירוט נרחב, תעשו בדיוק כמו מה שהכנתי לכם בעמודים הקודמים, כותרת שמציגה את הרכיב, ואז פירוט בנוגע לשימוש שעשיתם בו בפרויקט שלכם. ככל שזה יהיה ברור יותר לבוחן, כך טוב יותר.

בין הרכיבים המתקדמים נמנים: `SurfaceView`, `View`, `GoogleMaps` (לפחות הצגת מיקום ע"י כתובת, נקודות אורך רוחב), `מאזיני תנועה`, `NFC`, `Bluetooth`, `Threads`, פיתוח ושימוש ב `API` בצד שרת (אפשר בכל שפת תכנות), בינה מלאכותית, רשתות מחשבים, למידת מכונה, או טכנולוגיה חדשה (כגון עיבוד תמונה). `Camera & Gallery`, `AsyncTask`, `Handler`, `Microphone`, `GPS`, `SpeechToText`, `TextToSpeech`, `CountDownTimer`.

מידה ואין הרבה מה לפרט על הרכיב, אפשר להשאירו בעמוד זה ולכתוב:

- סוג הרכיב
- שימוש, מה אנחנו עושים איתו באפליקציה
- שם הקובץ בו נמצא

סוג הרכיב: `Handler`

שימוש: עדכון `UI Thread` מתוך `Thread` משני.

שם הקובץ בו נמצא: `LoginActivity.java`, `GameActivity.java`

סוג הרכיב: `Thread`

שימוש: תפעול שעון עצר לזמן של כל שחקן.

שם הקובץ בו נמצא: `GameActivity.java`





ביבליוגרפיה

שם הספר: "Android פיתוח אפליקציות"

שם המחבר: רועי אמיגה

קישור: https://drive.google.com/file/d/11grUaJ8cBru3KqFeWC7M2jHp_YL0kqTE/view?usp=sharing

Android Color Picker

ייעוד התוכנה: ספרייה בעלת דיאלוג המאפשר בחירת צבע.

יצרן: Randy Sugianto

גירסא: 2.0.1

קישור לGithub: <https://github.com/yukuku/ambilwarna>

ChatGPT: Conversational AI Language Model

ייעוד התוכנה: מודל שפת AI.

יצרן: OpenAI

גירסא: GPT-3.5

קישור: <https://openai.com/blog/chatgpt>

DALL-E

ייעוד התוכנה: בינה מלאכותית היוצרת תמונות מתיאורים טקסטואליים.

יצרן: OpenAI

גירסא: DALL. E 2

קישור: <https://openai.com/product/dall-e-2>



Midjourney

ייעוד התוכנה: בינה מלאכותית היוצרת תמונות מתיאורים טקסטואליים.

יצרן: Midjourney.

גירסא: Midjourney V4

קישור: <https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F>

