# ניהול הפרויקט

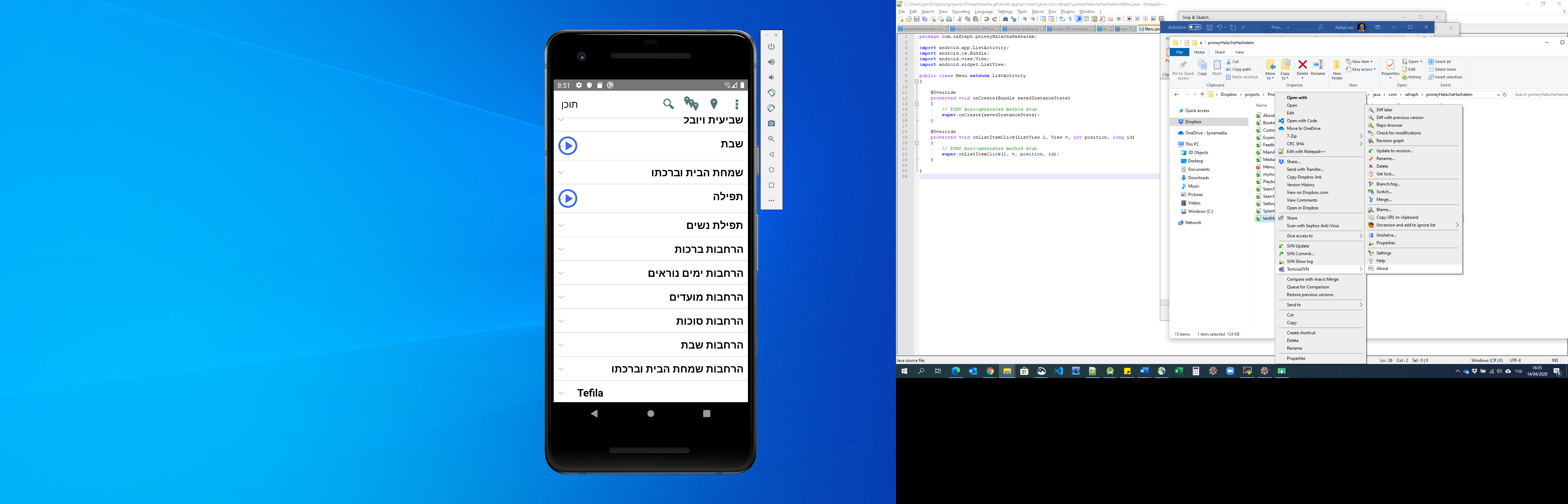
אפליקציית פניני הלכה מנוהלת ב-SVN ב-github. מי שמכיר רק git יוכל ללמוד יחסית בקלות SVN. הוא מאוד אינטואיטיבי והרבה יותר פשוט מ-git.

לאחר שמקבלים הרשאה ב-github ממנהל הפרוייקט, צריך ליצור PAT (personal access token). כפי שמוסבר כאן:

<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

ואז כשרוצים לעשות פעולות ב-SVN כמו למשל commit נקבל מסך של

אפשר להשתמש ב-SVN ב-command line אבל הכי פשוט לעבוד עם tortoisesvn. מורידים את הגירסה האחרונה מ- <https://tortoisesvn.net/downloads.html> ואז קל מאוד לעבוד עם SVN. ע"י מקש ימני של העכבר ניתן לראות את כל התפריט של SVN.



כל קובץ שיש לו אייקון ירוק, לדוגמה  זה אומר שמאז שהורדנו אותו מהשרת לא שינינו אותו. ואם יש לו אייקון אדום, לדוגמה  זה אומר ששינינו אותו. כדי לראות מה השינוי, מקש ימני על הקובץ ולחיצה על Diff. אם יש שינוי בשרת לא נראה את זה באייקון. כדי לראות את זה נלחץ מקש ימני על הקובץ או הספריה ואז show log ונוכל לראות מודגש את הגירסה שאנחנו משתמשים בה. ואם יש מעל המודגש עוד שורות זה אומר שיש שינויים בשרת וכדי לקחת אותם נעשה update.

בגדול אלו הן הפקודות הבסיסיות שנצרכות:

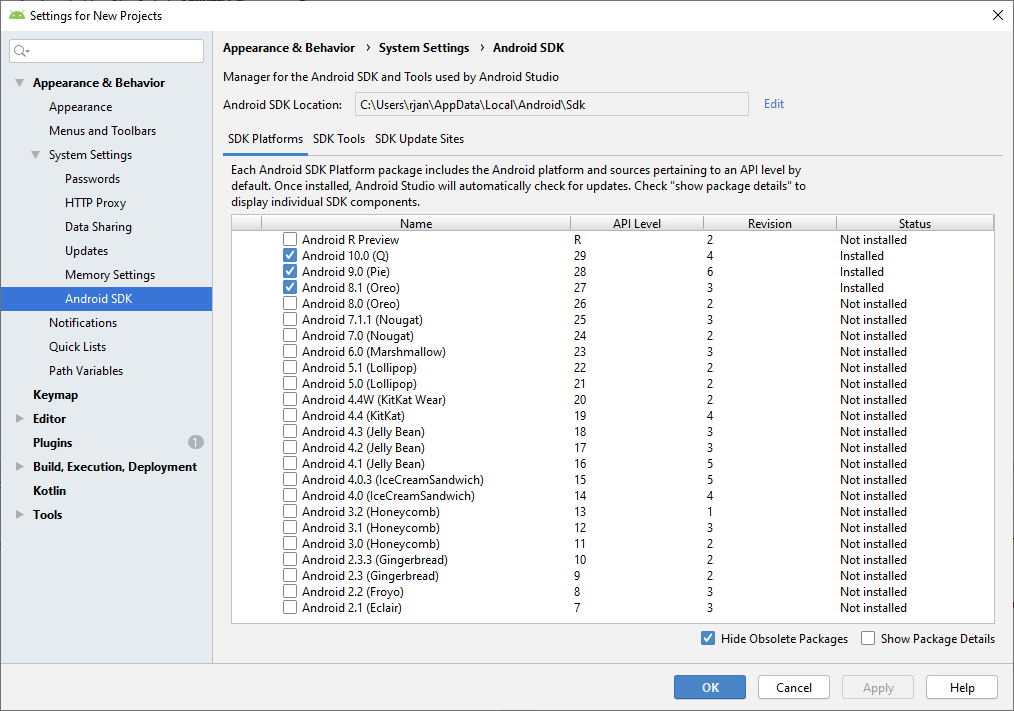
1. Checkout – הורדת הפרויקט מ-github
2. Commit – הכנסה של שינויים ל- github(דומה ל-Push של git)
3. Show log – כדי לראות רשימה של כל ה-commit שנעשו עד היום
4. Update – הורדת כל השינויים שנעשו בשרת אל המחשב שלך
5. Check for modifications – כדי לקבל רשימה של כל הקבצים ששינת ועדיין לא הוכנסו לשרת
6. Add – הוספה של קובץ

כמובן שגם מחיקה של קובץ או שינוי שם צריכים להעשות דרך SVN ולא בצורה רגילה

# פיתוח

הפיתוח מתבצע ב-Android Studio. כשמורידים אותו הוא מוריד כמעט את כל מה שצריך. חפשו באינטרנט Android Studio והתקינו. לכו לעשות קפה ואולי שתיים כי זה לוקח זמן.

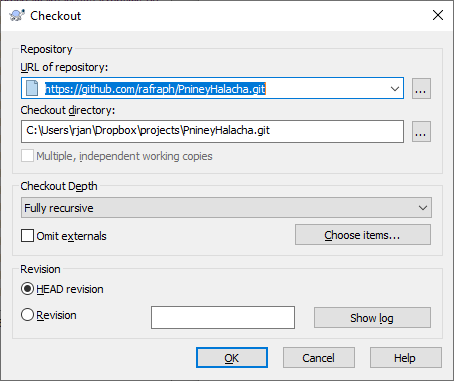
לאחר שהוא ירד פתח אותו וגש ל- Tools 🡪 SDK manager שם ניתן להוריד גרסאות שונות של SDK של אנדרואיד. תוריד שתיים או שלוש גרסאות אחרונות. נכון לעכשיו הגרסאות האחרונות הן 10,9,8.1.



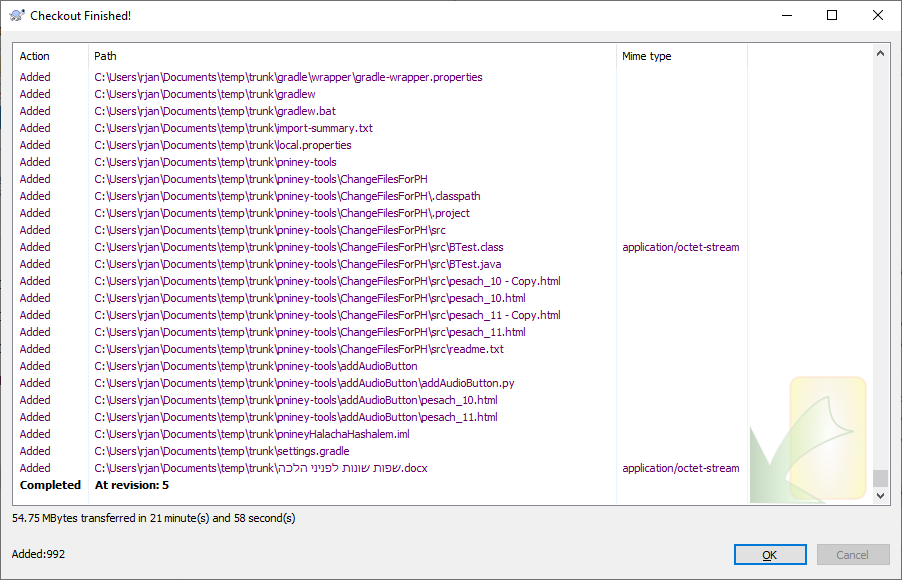
מדיי פעם יש קצת בעיות עם Android Studio בעיקר בפעם הראשונה או אחרי הרבה זמן שלא משתמשים בו. מנסיון קל מאוד למצוא פתרונות לבעיות האלו באינטרנט. סומך עליכם.

# הורדת הפרויקט

יוצרים תיקייה בשם PnineyHalacha ובתוכה לוחצים על מקש ימני ו-Svn Checkout. בחלון שנפתח מכניסים בשורה הראשונה: <https://github.com/rafraph/PnineyHalacha.git> ולוחצים על OK.

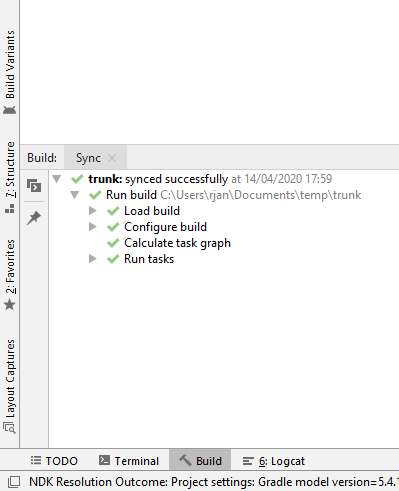


כשתהליך ההורדה מסתיים זה נראה משהו כזה:



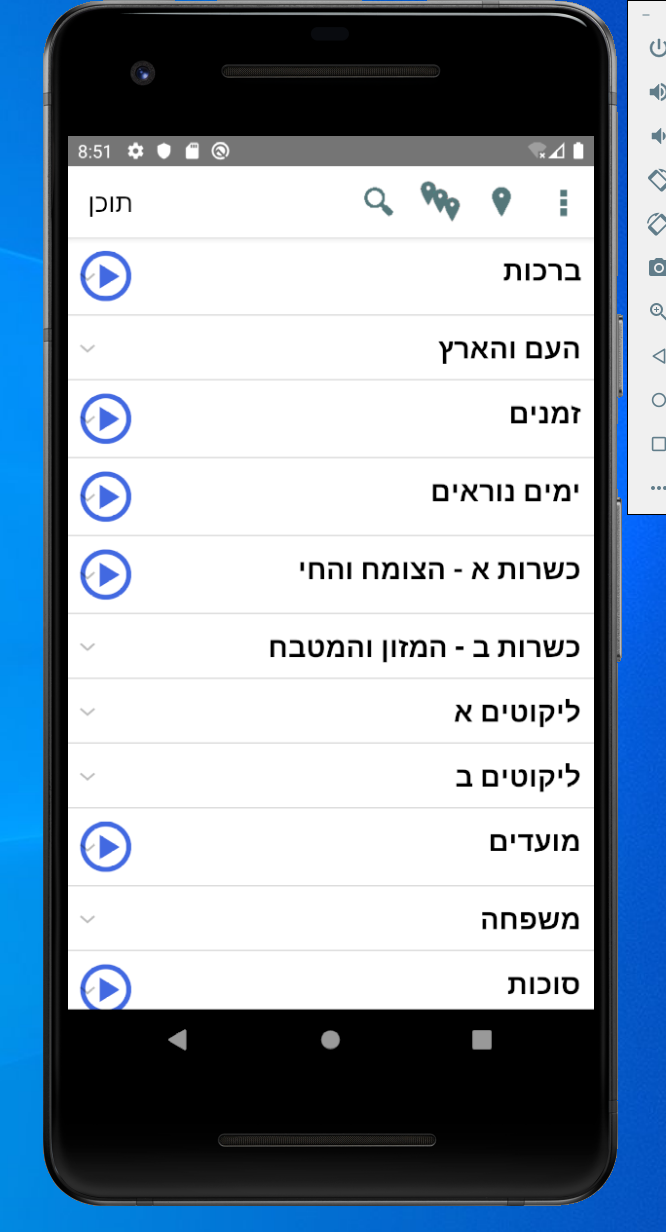
# הפעלה ראשונית

נפתח את Android Studio ונלחץ על File🡪New🡪Import project נחפש את הספריה PnineyHalacha שיצרנו ונבחר בספריה trunk שנמצאת בה ונלחץ עך OK. הפרוייקט יפתח ואוטומטית יתחיל תהליך קומפילציה. אם הוא יסתיים בצורה מוצלחת נראה בחלק התחתון של Android Studio משהו כזה:



אם הוא לא מתקמפל אוטומטית אז נלחץ על Build🡪Make project.

נוודא שהאפליקציה עובדת על האמולטור ע"י לחיצה על Run🡪Run app אם זה עובד יפתח לנו האמולטור עם האפליקציה וזה יראה כך:



# מבנה הפרוייקט

קבצי ה-Java נמצאים ב- trunk\app\src\main\java\com\rafraph\pnineyHalachaHashalem

להלן הסבר קצר על כל קובץ:

**MainActivity.java** – המסך הראשי שבו יש את רשימת הספרים והפרקים שלהם

**textMain.java** – המסך שמכיל את הטקסט

**About.java** – קובץ קטן שאחראי על אחד התפריטים

**BookmarkActivity.java** – אחראי על הסימניות

**CustomSpinnerAdapter.java** – קובץ קטן שקשור למערכת הסימניות

**ExpandableListAdapter.java** – מאפשר את הלחיצה על שם ספר שפותחת רשימה פנימית עם שמות הפרקים

**Feedback.java** – מסך לשליחת פידבקים

**MediaPlayerService.java** – ה-serivce של הנגן לצורך הקראת ההלכות

**PlaybackStatus.java** , **myAudio.java** – קשורים לנגן ההקראות

**SearchableActivity.java** – אחראי על החיפוש באפליקציה

**SearchHelp.java** – מסך עם הסבר על תוצאות החיפוש

**Settings.java** – אחראי על ההגדרות באפליקציה

**Splash.java** – אחראי על מסך ה-splash שמופיע בפתיחת האפליקציה

**Menu.java** – קובץ קטן שקשור לתפריטים

הקבצים האחראים על הגרפיקה נמצאים ב- trunk\app\src\main\res בעיקר בספריית layout.

# שינוי טקסט בספרים

לפעמים יש צורך לשנות טקסט בספרים. או כי יש טעות למשל ירידת שורה מיותרת או שהרב מעדכן את התוכן. כל הטקסט שמור כקבצי HTML והוא נמצא ב- trunk\app\src\main\assets. לכל פרק יש קובץ עם שם ברור למשל עבור ברכות פרק ג הקובץ הוא brachot\_3.html.

האיורים שיש בספרים נמצאים ב- trunk\app\src\main\assets\images.

אם השינוי הוא פשוט, למשל טעות במילה, אפשר לגשת ישירות לאותו פרק שבו הטעות ולתקן.

אם יש צורך בשינוי של כל הפרק אז התהליך מורכב יותר. להלן הצעדים הנצרכים.

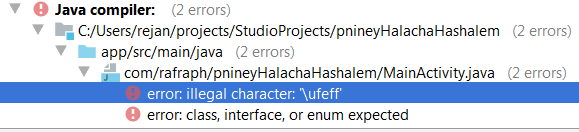
1. מקבלים קובץ WORD של אותו פרק שרוצים לשנות. הקובץ כתוב בצורה עיצובית נכונה, למשל כל כותרת מוגדרת ככותרת וכל הערת שוליים מוגדרת כהערת שוליים. מעתיקים את כל התוכן ע"י ctrl+a ומדביקים את הכל באתר <https://word2cleanhtml.com/> האתר הזה ממיר קובץ WORD ל-HTML בצורה נקייה מאוד. לוחצים על CONVERT. מעתיקים את מה שקיבלנו ע"י לחיצה על Copy cleaned HTML to clipboard. פותחים את הקובץ של הפרק שרצינו לעדכן ב- notepad++ או כל editor אחר ומדביקים.



1. עכשיו יש צורך להשתמש בסקריפט שכתבנו בג'אווה שמסגנן את הטקסט (כל סעיף יהיה לו סימניה (כדי שיהיה אפשר להגיע אליו דרך התפריט), וכל הערת שוליים הופכת לכפתור, וכו'). הסקריפט נמצא ב- **trunk\pniney-tools\ChangeFilesForPH\src** מעתיקים למיקום הזה את הקובץ שרוצים לשנות (שומרים את הקובץ המקורי לשם השוואה בסיום הסקריפט) ופועלים לפי ההוראות שנמצאות בקובץ **readme.txt** שנמצא באותו מיקום.
2. לאחר שהסקריפט ChangeFilesForPH סיים נשווה את מה שקיבלנו עם הקובץ המקורי. אנחנו אמורים לראות הוספת סימניות לשמות הסעיפים (זה נראה משהו כזה <a name="1"></a>) והוספת כפתורים במיקומים של הערות השוליים. זה בגדול השינויים. אם זה נראה בסדר פותחים את זה בדפדפן כדי לראות בעיניים שזה בסדר ואין ג'יבריש או טקסט לא טוב.
3. אם זה פרק שעדיין אין לו הקראה (של הרב יונדב) אז סיימנו
4. אם יש לפרק הקראה צריך להריץ סקריפט נוסף שמוסיף את כפתורי ה-Play כדי לנגן את ההקראות. הסקרויפט כתוב בפייתון והוא נמצא ב- **trunk\pniney-tools\addAudioButton** הוא נקרא **addAudioButton.py**. מעתיקים את הקבצים לאותו מיקום ואז פשוט פותחים command line ומריצים אותו: python addAudioButton.py. או בכל דרך הרצה המועדפת עליך.
5. לאחר שהסקריפט סיים נשווה את מה שקיבלנו לקובץ המקורי. אנחנו אמורים לראות רק הוספת כפתורים עבור ההקראה. הכפתורים מתווספים לכותרת הסעיף ונראים משהו כזה:

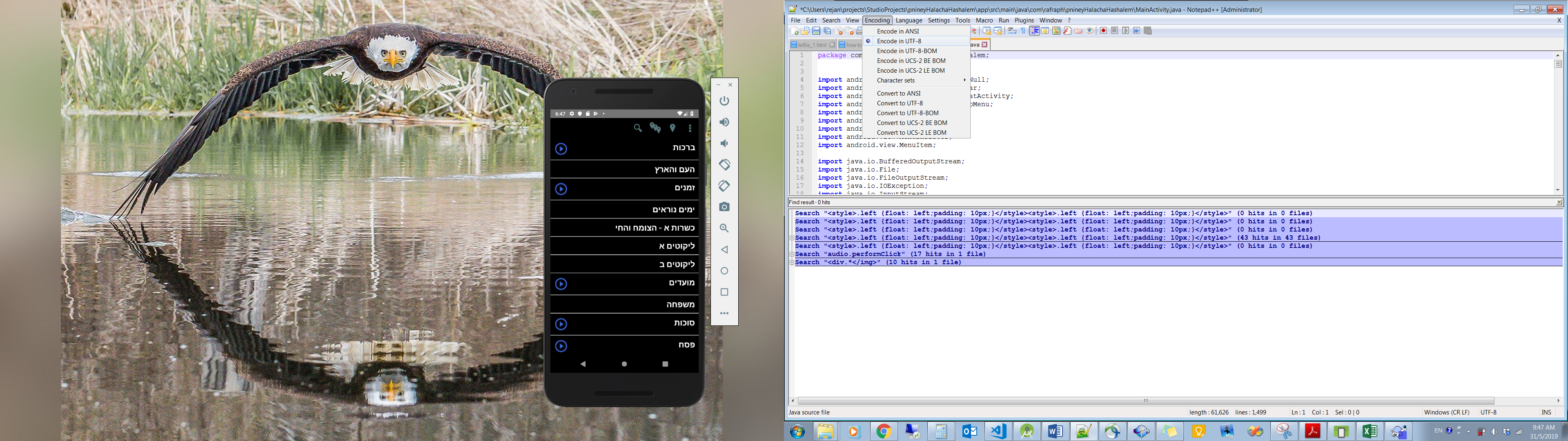
<div class ="left"><img type="button" id="1" onclick="audio.performClick(id);" src="./video-play-3-128.png" width="36px"; height="36px";></img></div>

# Errors and solutions

1. If you get this error: 

Open the problematic file (in this case MainActivity.java) in notepad++.

Encoding 🡪 Encode in UTF-8



Save the file

1. If you have problems like crash without explanation in android studio go to cmd and write:

adb logcat --buffer=crash

See this: <https://stackoverflow.com/questions/3643395/how-to-get-android-crash-logs>

If adb is not recognized you need to insert its path to the environment variables. See this: <https://stackoverflow.com/questions/20564514/adb-is-not-recognized-as-an-internal-or-external-command-operable-program-or>