

פיתוח אפליקציות

שיעור מספר 1

שיעור 1 - שאלות ותשובות + תרגול

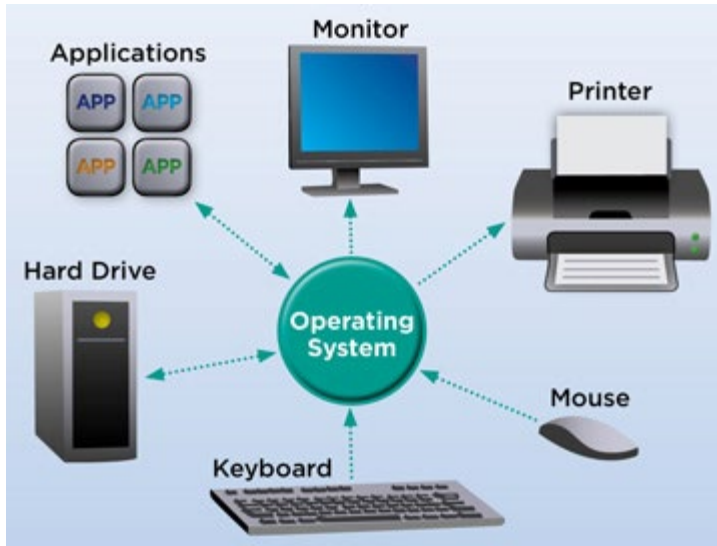
1. היכרות
2. מה זה מערכת ההפעלה?
3. מה זה מערכת ההפעלה אנדרואיד?
4. מה זה אפליקציה?
5. מה עושים במהלך הקורס?
6. מה זה APP INVERTOR ?
7. פיתוח אפליקציה הראשונה - HELLO WORLD

נעים להכיר

1. שם
2. היכרות עם מערכת הפעלה אנדרואיד
3. היכרות עם GOOGLE
4. היכרות עם שפת תכנות כלשהי
5. היכרות עם פיתוח אתרים

מערכת הפעלה

- ▶ מערכת הפעלה באנגלית - OPERATION SYSTEM
- ▶ התוכנה אשר מנהלת את משאבי החומרה והתוכנה במחשב.
- ▶ בנוסף, מערכת ההפעלה מספקת את התשתית הנחוצה להרצה של יישומי ההפעלה, המתבצע עם הדלקת המחשב, קרוי אתחול.
- ▶ מערכת ההפעלה מספקת שלושה ממשקים:
 - ▶ ממשק משתמש (User Interface)
 - ▶ ממשק עבור החומרה על ידי מנהלי התקנים
 - ▶ ממשק תכנות היישומים API -



▶ מערכת ההפעלה היא רכיב חיוני בכל מחשב!!!!

מערכת ההפעלה

ניתן למנות שלושה תפקידים עיקריים של מערכת ההפעלה

- ▶ הקצאת משאבי החומרה
- ▶ תזמון פעולות רכיבי החומרה ומרכיבי התוכנה
- ▶ העמדת תשתית משותפת ומסגרת מאורגנת של ממשק ושירותים למשתמש ולחבילות התוכנה



סוגי מערכות הפעלה

3 מערכות הפעלה הפעילות והעיקריות הן

WINDOWS ►

LINUX ►

IOS ►

ANDROID (מבוססת LINUX) ►

מערכת הפעלה WINDOWS

- ▶ מה אתם יודעים על WINDOWS?
- ▶ פותחה על ידי חברת מייקרוסופט
- ▶ גרסה ראשונה בשנת 1985
- ▶ גרסה פעילה נוכחית WINDOWS 10
- ▶ מופצת בתשלום



Windows 10

מערכת הפעלה LINUX

- ▶ מה אתם יודעים על LINUX?
- ▶ מערכת הפעלה פתוחה - מבוססת על OPEN SOURCE - קוד פתוח
- ▶ ניתן להוריד בחינם
- ▶ ניתנת לפיתוח לתכניתנים בכל העולם
- ▶ הרבה ארגונים בעולם הורידו את הקוד של לינוקס בחינם והרחיבו ושינו אותו והוציאו LINUXים משלהם - נקרא "הפצה של לינוקס"
- ▶ בעצם אנחנו משתמשים עכשיו בלינוקסים המופצים.
- ▶ בין ההפצות הכי מפורסמות של לינוקס ניתן למנות את REDHAT , UBUNTU , DEBIAN
- ▶ אבל יש עוד המון (תראו את התמונה!!!)
- ▶ מערכת הפעלה ANDROID גם היא בעצם המשך של לינוקס



מערכת הפעלה ANDROID

- ▶ מערכת הפעלה המיועדת לסמארטפונים, טאבלטים, טלוויזיות חכמות, שעונים חכמים ולמכונות
- ▶ אנדרואיד היא מערכת ההפעלה המבוססת LINUX
- ▶ היא מופצת על ידי חברת גוגל GOOGLE (כי גוגל זה לא רק חיפוש (:))
- ▶ אנדרואיד מחזיקה כ-70 אחוז מכלל שוק של כל הטלפונים החכמים!!



מערכת הפעלה ANDROID

- ▶ Android .inc נרכשה ע"י Google ב 2005
- ▶ מערכת הפעלה מבוססת לינוקס למכשירים סלולריים
- ▶ תכנות אפליקציות באמצעות Java
- ▶ ב 2007 נחשף לציבור
- ▶ רובו קוד פתוח תחת רישיון apache
- ▶ ברבעון הראשון של 2010 מכירות של מערכת ההפעלה android ל smart phones בארה"ב היו

מערכת הפעלה ANDROID

- ▶ כמו לכל מערכת ההפעלה, לאנדרואיד יש חיים משלה.
- ▶ כל כמה זמן, בוחרת חברת GOOGLE להוציא גרסה חדשה שלה.
- ▶ לגרסה החדשה מועברים יכולות החדשות שפותחו זה עתה והתקלות אשר תוקנו בגרסאות הקודמות של האפליקציה
- ▶ להלן שמות של הגרסאות של המערכת ההפעלה

ANDROID VERSION UNTIL NOW !!!



מערכת הפעלה ANDROID

- ▶ מערכת ההפעלה אנדרואיד מאפשרת פיתוח בשפת תכנות JAVA
- ▶ פיתוח אפליקציות כולל מתן אפשרות גישה לכל משאבי המחשב
- ▶ איזה משאבי מחשב אתם מכירים??

מערכת הפעלה ANDROID

משאבי אנדרואיד ►

מצלמה ►

מיקרופון ►

קבצים ►

מקלדת ►

ספר טלפונים ►

מסמכים ►

ועוד ועוד.... ►

מערכת הפעלה ANDROID

- ▶ כל אפליקציה רצה בתוך תהליך משלה
- ▶ אפליקציה אשר מעמיסה על המעבד לא תחסום את התהליכים הקריטיים אחרים כגון מענה לטלפון
- ▶ מערכת Android אחראית להפעלתם ועצירתם של תהליכים כנדרש להרצתן של אפליקציות
- ▶ יתכן שתהליכים יהרגו (ואח"כ ישוחזרו) על מנת לפנות משאבים לתהליכים אחרים

מערכת הפעלה ANDROID

- ▶ מערכת ההפעלה אנדרואיד היא מערכת לינוקס מרובת משתמשים שבה כל יישום הוא משתמש אחר.
- ▶ כברירת מחדל, המערכת מקצה לכל יישום מזהה משתמש ייחודי של לינוקס (המזהה משמש רק על-ידי המערכת ואינו ידוע ליישום).

מערכת הפעלה ANDROID

- ▶ המערכת מגדירה הרשאות עבור כל הקבצים ביישום כך שרק מזהה המשתמש שהוקצה ליישום זה יוכל לגשת אליהם.
- ▶ לכל תהליך יש מחשב וירטואלי משלו (VM), כך קוד אפליקציה פועל בבידוד מיישומים אחרים.
- ▶ כברירת מחדל, כל יישום פועל בתהליך לינוקס משלה. מערכת אנדרואיד מתחיל את התהליך כאשר כל הרכיבים של האפליקציה צריך להתבצע, ולאחר מכן מכבה את התהליך כאשר זה כבר לא נחוץ או כאשר המערכת חייבת לשחזר את הזיכרון עבור יישומים אחרים.

מערכת הפעלה ANDROID

- ▶ נניח שמאפליקציית Home פתחנו Gmail
- ▶ בחרנו מכתב,הצגנו אותו וראינו בתוכו קישור
- ▶ לחצנו על הקישור והדפדפן נפתח בעמוד כלשהו
- ▶ בעמוד יש קישור למפה שלחצנו עליו
- ▶ האפליקציה של תצוגת המפות נפתחה
- ▶ כעת אנו רוצים לחזור מסכים אחורה...
- ▶ וכך הלאה - כל כפתור בסולארי שלנו הוא אפליקציה!!

APP INVERTOR

- ▶ אפליקציה ואתר אשר פותחו בעבר על ידי GOOGLE
- ▶ כרגע מתוחזקת על ידי MIT - אוניברסיטה אמריקאית
- ▶ מאפשר לכל אחד - כולל אנשים אשר אינם יודעים לפתח, ליצור אפליקציות אנדרואיד.
- ▶ כרגע מתוחזקת תחת השם MIT APP INVERTOR

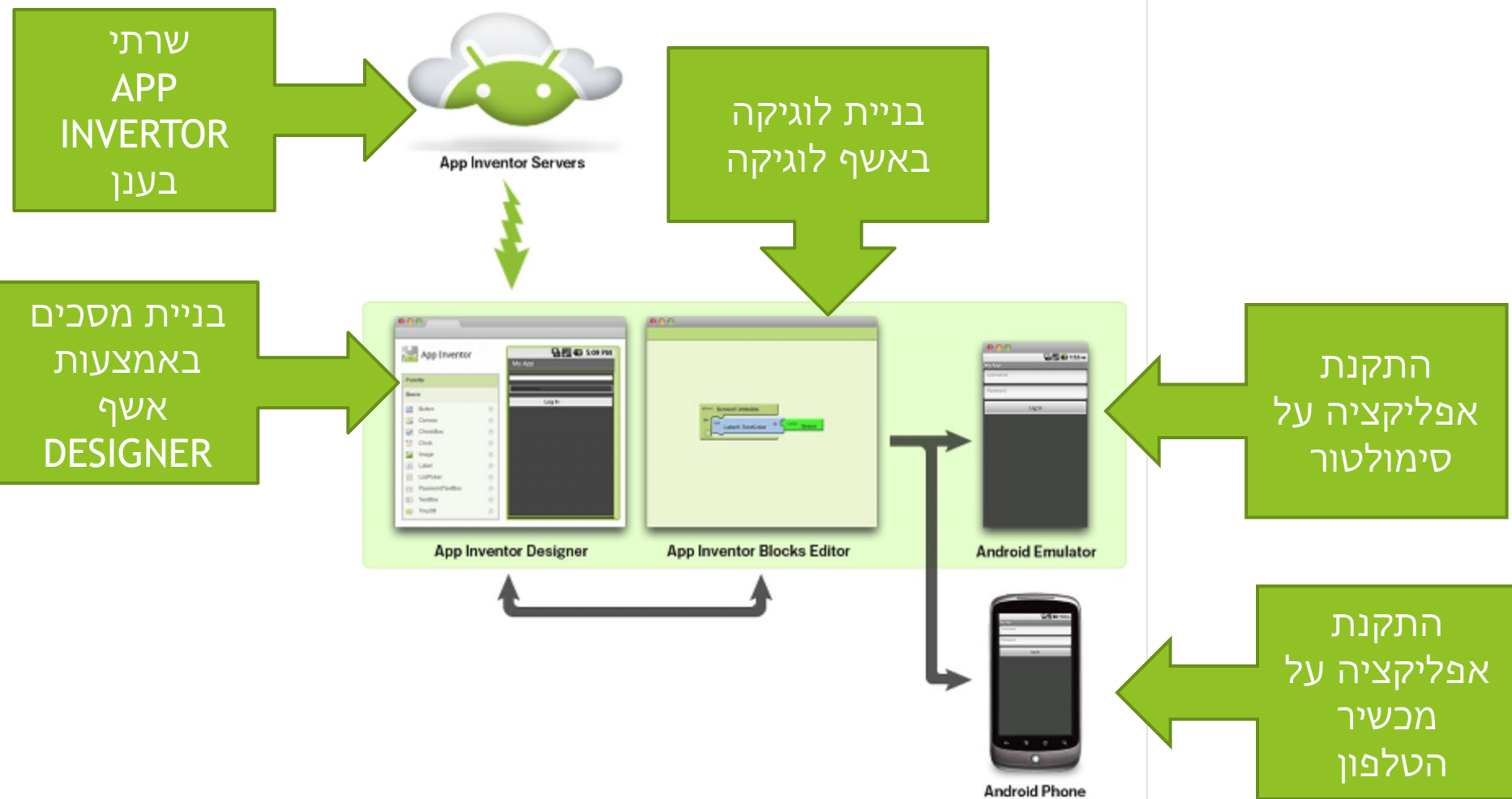


APP INVERTOR - מדוע פיתוח כל כך קל?

- ▶ **אין תחביר** . שפת הבלוקים מבטלת את הצורך לזכור ולהקליד קוד
- ▶ **הכל נמצא ממש מולך** . רכיבים ופונקציות נמצאים במגירות . פשוט למצוא, לגרור ולשחרר
- ▶ **אירועים ברמה העליונה** . "כאשר זה קורה, האפליקציה עושה את זה" הוא המודל הרעיוני הנכון.
- ▶ **רכיבים ברמה גבוהה** . צוות APP INVERTORE בנה ספרייה גדולה עם מטרה פשוטה.
- ▶ **רק לגרור כמה בלוקים** . אתה לא יכול לעשות דברים שלא הגיוניים
- ▶ **קונקרטיות** . פחות מופשט מהרבה שפות.

APP INVERTOR - מה אפשר לפתח?

- ▶ אפליקציות חינוכיות - כולל וידאו, תמונות, טקסט לדיבור
- ▶ אפליקציות מבוססות מיקום
- ▶ משחקים
- ▶ יישומים מותאמי אינטרנט
- ▶ אפליקציות אישיות



הפעלה ראשונה של APP INVERTOR ופיתוח האפליקציה הראשונה

- ▶ עכשיו נפתח את האפליקציה הראשונה שלנו... מתרגשים???
- ▶ אפליקציה ראשונה אצל כל תוכניתן מתחיל נקראת HELLO WORLD.
- ▶ האפליקציה היא מאוד מאוד בסיסית ופשוטה ובעיקר פיתוח שלה , מטרתו לבצע בדיקה שכל הסביבה שלכם עובדת ואתם מבינים את הAPP INVERTOR

ההפעלה ראשונה של APP INVERTOR

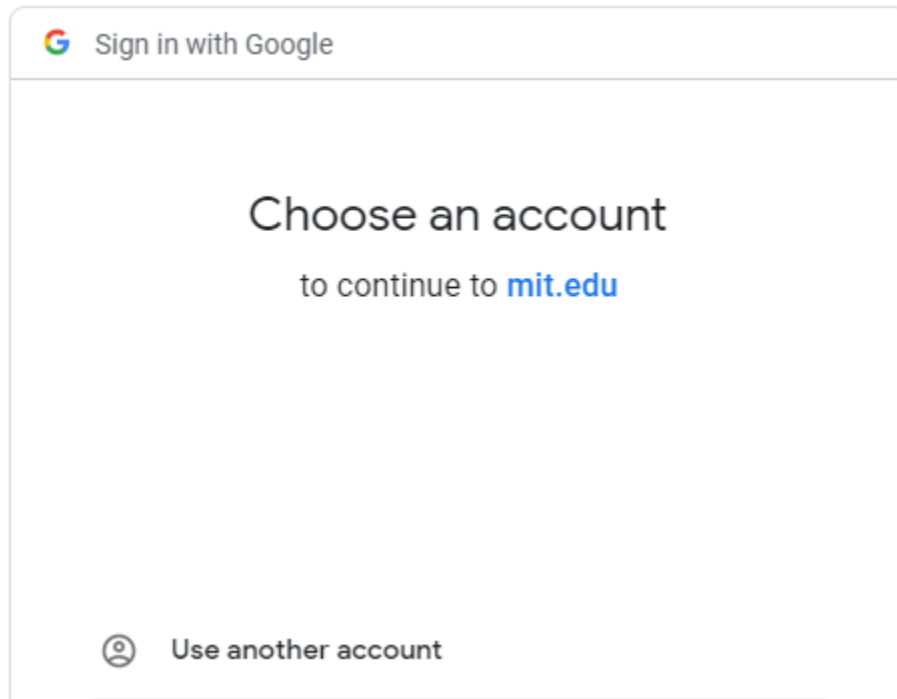
יש לוודא שהמחשב מחובר לאינטרנט ►

יש ליצור חשבון GMAIL ►

יש להתחבר ל ►

<http://ai2.appinventor.mit.edu/>

לאחר ההתחברות, תנובת להזנת חשבון GMAIL ►



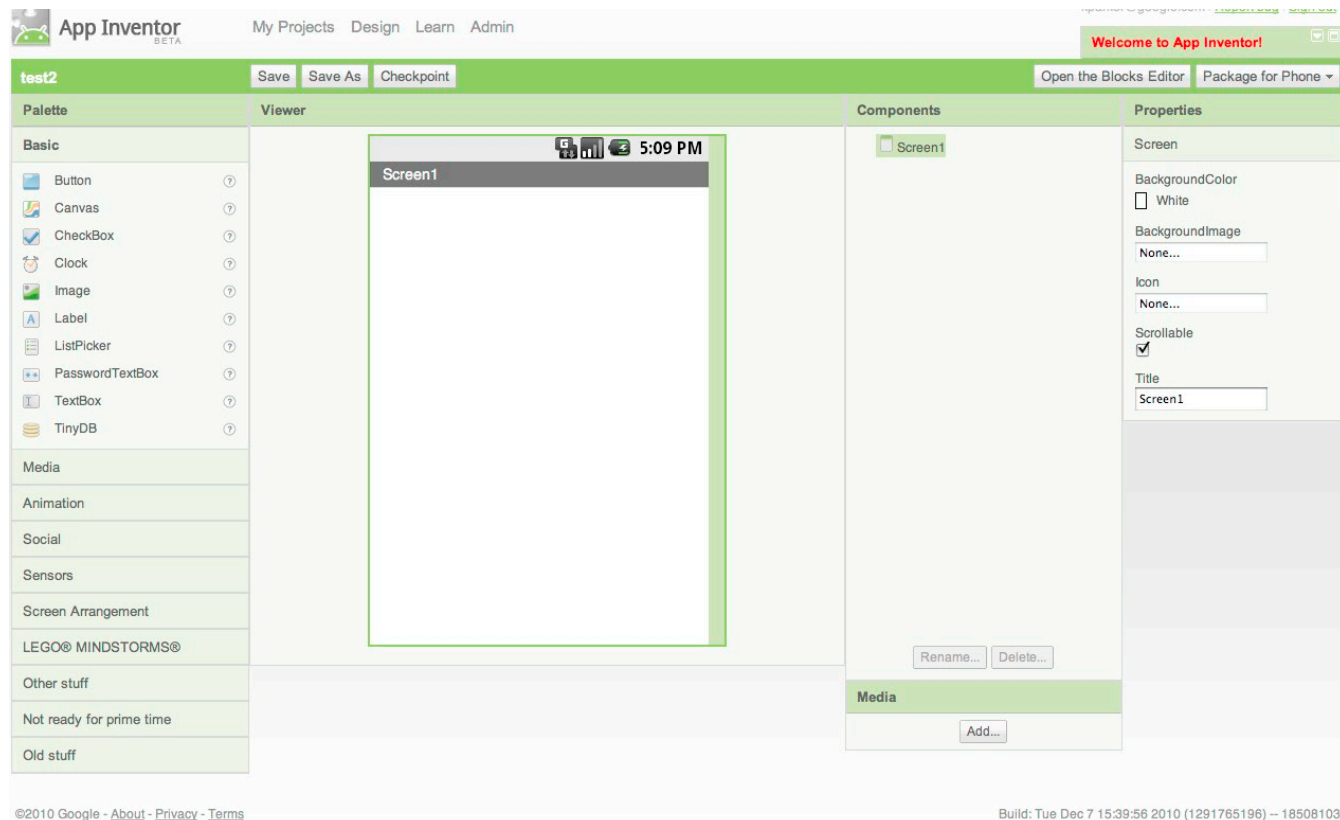
הפעלה הראשונה של APP INVERTOR

- ▶ תגיעו לחלון יצירת הפרויקט
- ▶ יש ללחוץ על כפתור NEW ליצירת פרויקט חדש
- ▶ הזינו את שם האפליקציה הראשונה שלכם - HELLO WORLD



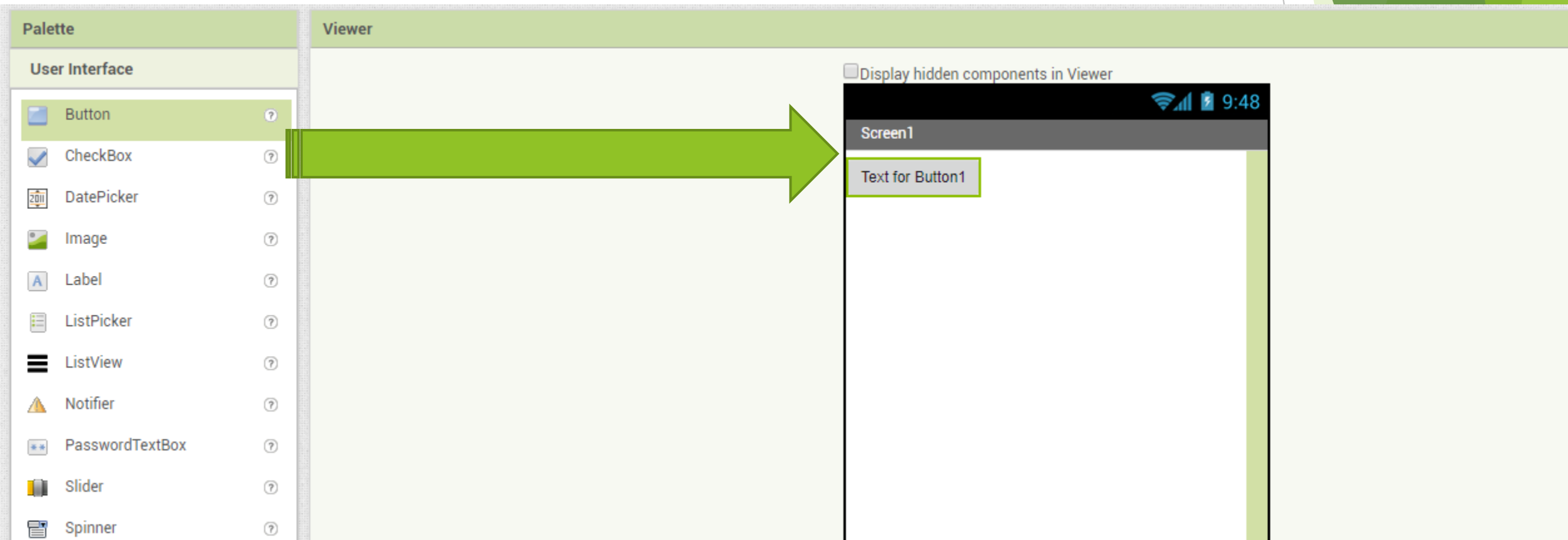
הפעלה הראשונה של APP INVERTOR

- ▶ הדפדפן יפתח לכם דף שנקרא DESIGNER.
- ▶ תפקידו של DESIGNER בפיתוח אפליקציות הוא לאפשר לכם לבנות את מסכי האפליקציה



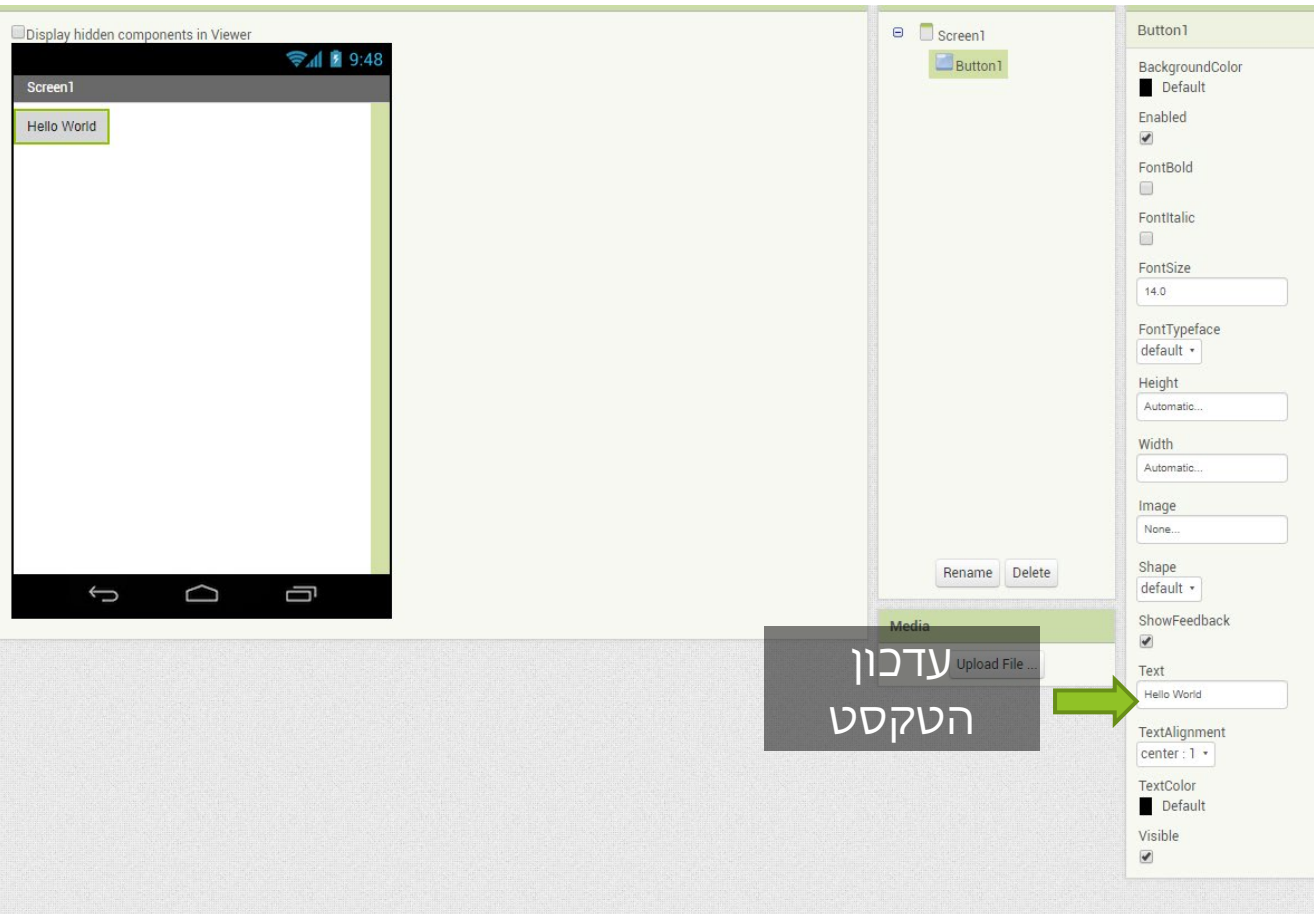
הפעלה הראשונה של APP INVERTOR

- ▶ הכפתורים מצדו השמאלי של המסך נקראים "פקדים"
- ▶ ניתן לגרור את הפקדים מצד שמאל לצד ימין ולמקם אותם על גבי המסך
- ▶ במקום שבו תמקמו את הכפתור, שם הוא יופיע באפליקציה



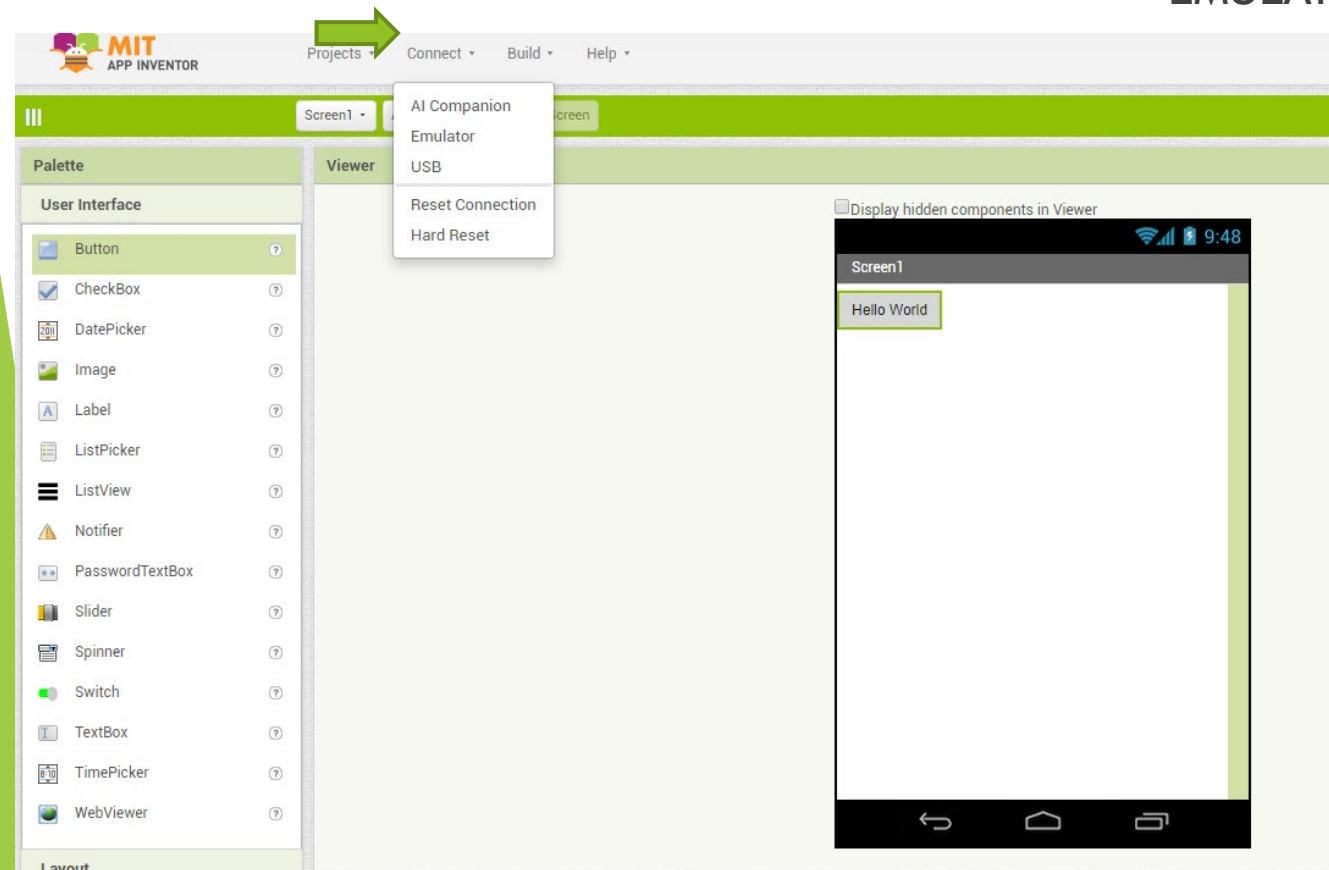
הפעלה הראשונה של APP INVERTOR

- ▶ לאחר העברת הפקד על גבי המסך, ניתן לעדכן את הטקסט אשר מופיע על גביו
- ▶ לצורך כך יש לבחור את הפקד ואז לעדכן את הטקסט שבו לטקסט הרצוי -
עדכנו את הטקסט לHELLO WORLD



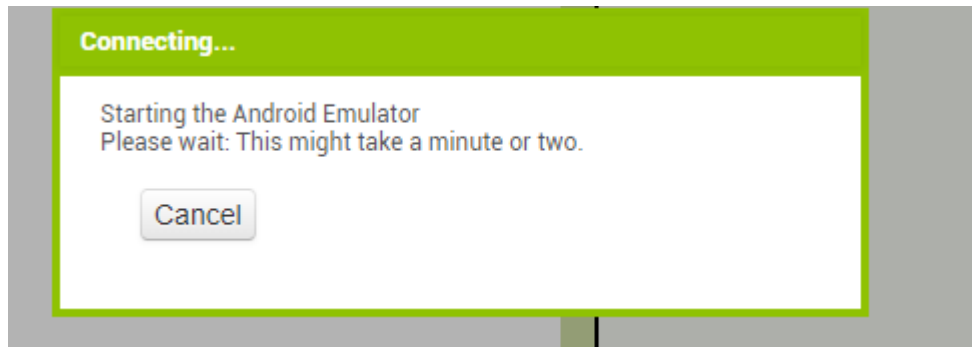
הפעלה ראשונה של APP INVERTOR

- ▶ נניח בשלב הראשון, שהאפליקציה שלנו לא עושה כלום - רק מציגה למשתמש דף ריק עם הכפתור HELLO WORLD
- ▶ כבר עכשיו ניתן להריץ את האפליקציה ולבדוק את עבודתה
- ▶ לצורך כך יש להפעיל את הסימולטור
- ▶ הפעלה של סימולטור מתבצעת על ידי בחירה בתפריט CONNECT באופציה EMULATOR



הפעלה ראשונה של APP INVERTOR

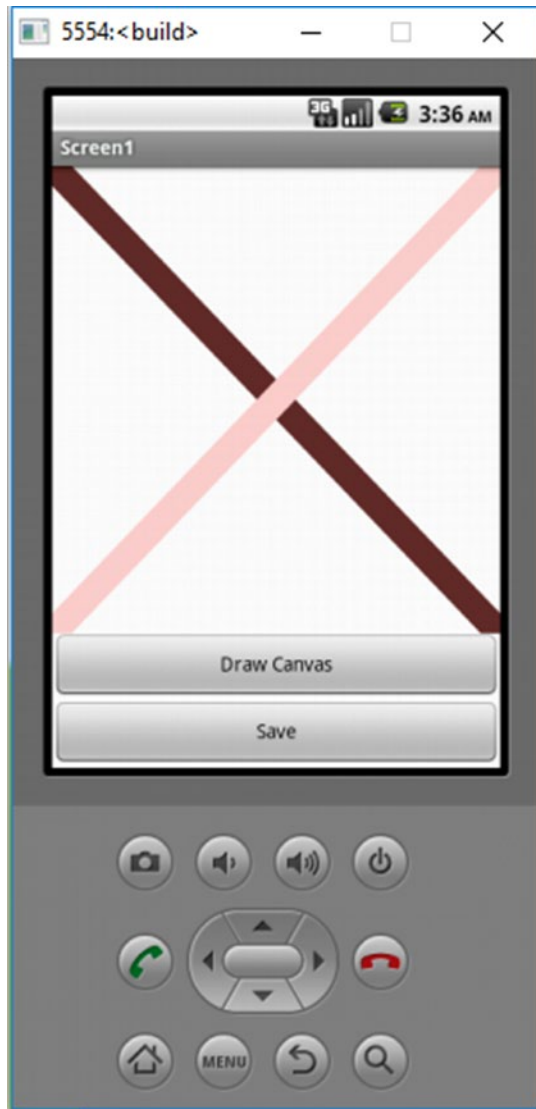
לאחר הפעלת EMULATOR, תופיע הודעה הבא ►



הפעלה ראשונה של APP INVERTOR

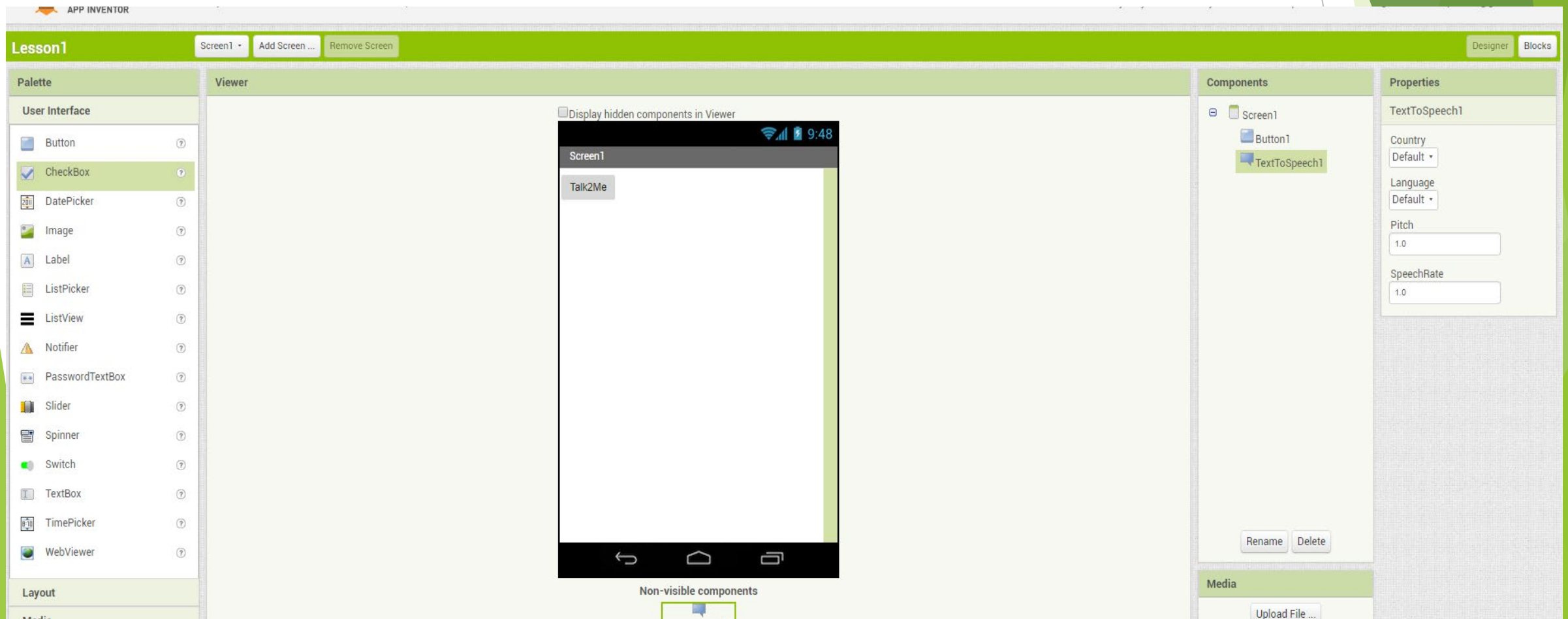
▶ לאחר מכן תפתח ה-EMULATOR

▶ וניתן יהיה לצפות בתוכנית שלנו רצה בו



פיתוח האפליקציה המדברת

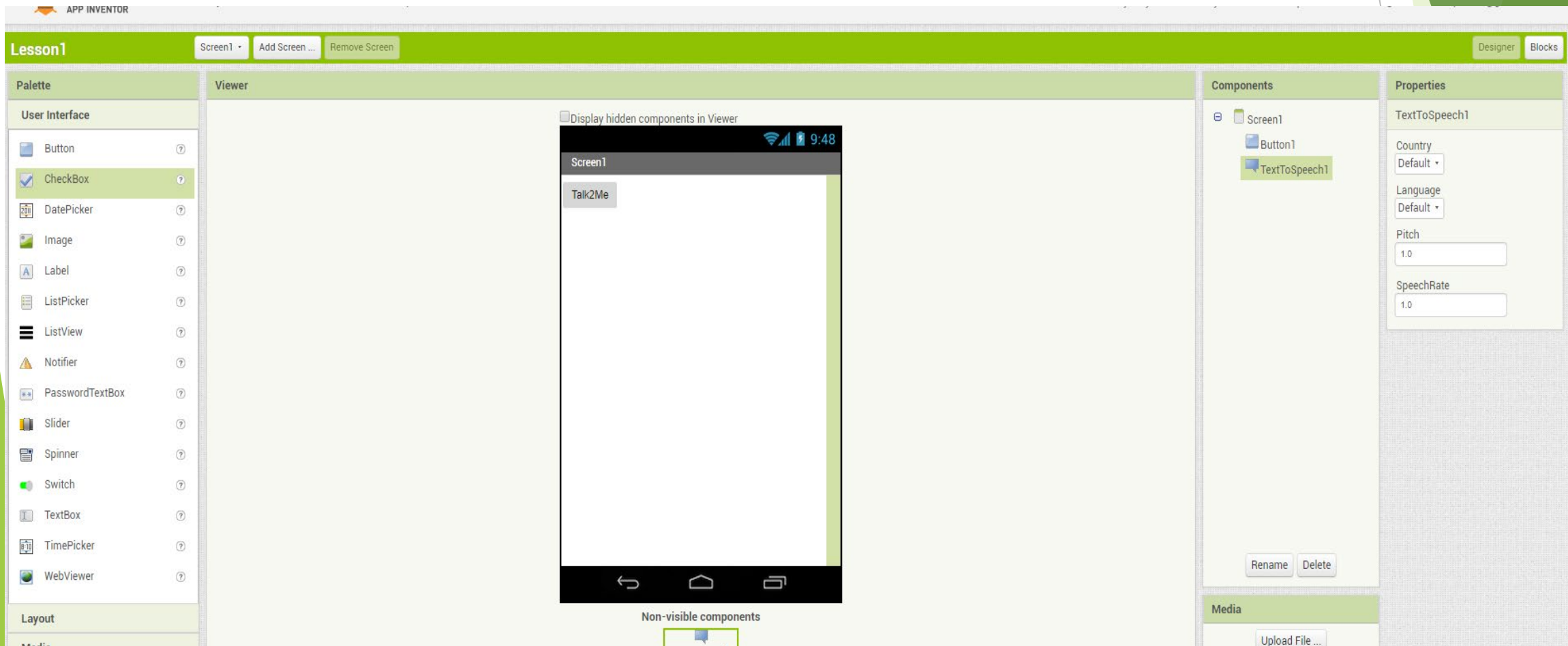
▶ פיתוח אפליקציה שמדברת בלחיצה על הכפתור (לא לשכוח להפעיל רמקול!!!)



פיתוח האפליקציה המדברת

▶ פיתוח האפליקציה שמדברת בלחיצה על הכפתור (לא לשכוח להפעיל רמקול!!!)

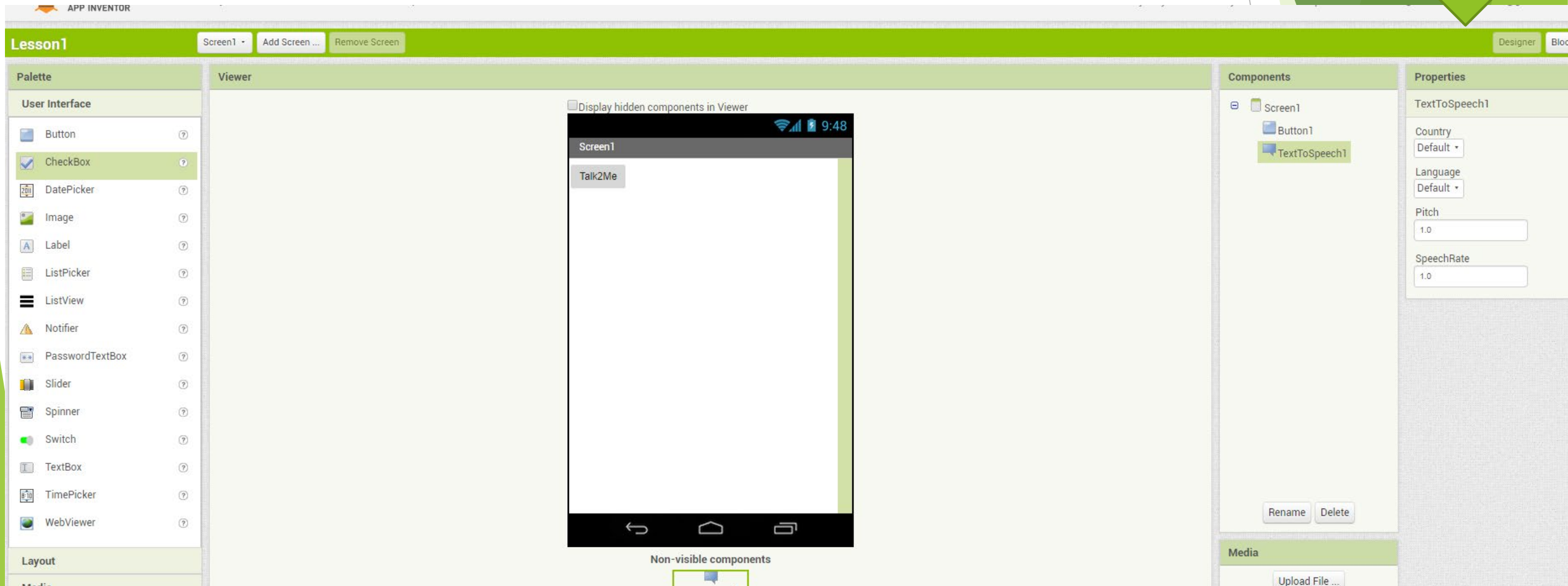
▶ שלב א - להוסיף כפתור ולשנות לו שם לME2TALK - דבר אלי



פיתוח אפליקציה מדברת

- ▶ פיתוח אפליקציה שמדברת בלחיצה על הכפתור (לא לשכוח להפעיל רמקול!!!)
- ▶ שלב א - להוסיף כפתור ולשנות לו שם לME2TALK - דבר אלי.
- ▶ שלב ב - ללחוץ על הכפתור הBLOCKS

כפתור
BLOCKS



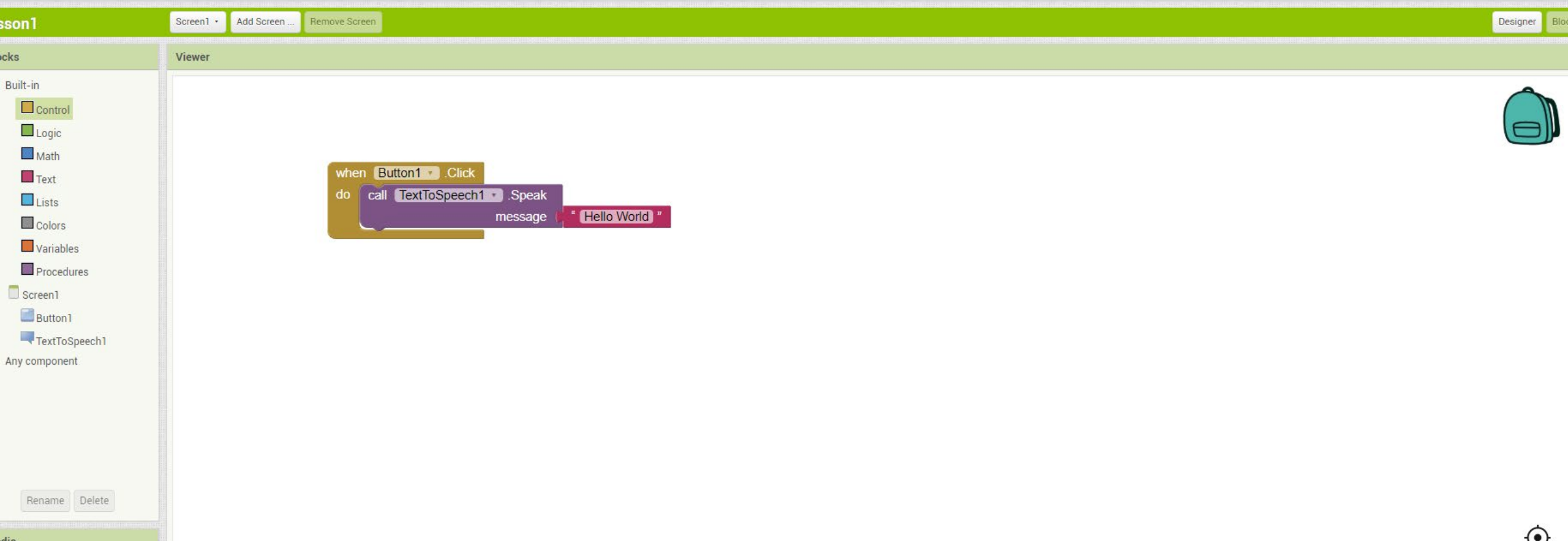
פיתוח אפליקציה מדברת

- ▶ בכפתור BLOCKS ניתן להוסיף לוגיקה פשוטה לתוכנית שלנו
- ▶ בשלב זה פשוט תנסו להעתיק את פתרון על ידי גרירה



פיתוח אפליקציה מדברת

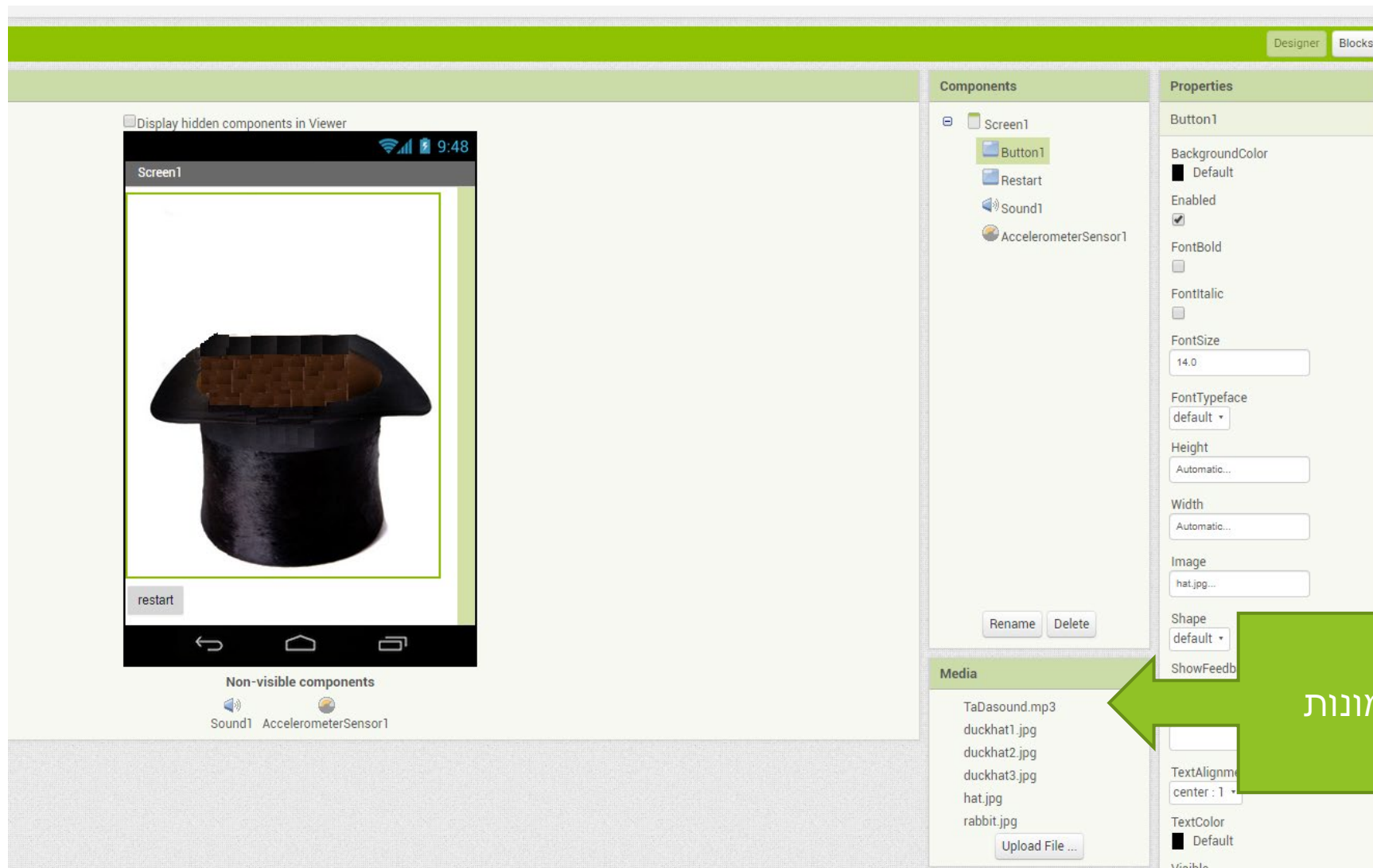
הפעלת את התוכנית בדרך הרגילה (על ידי אמולטור) ►



פתיחת אפליקציה קיימת - MAGIC

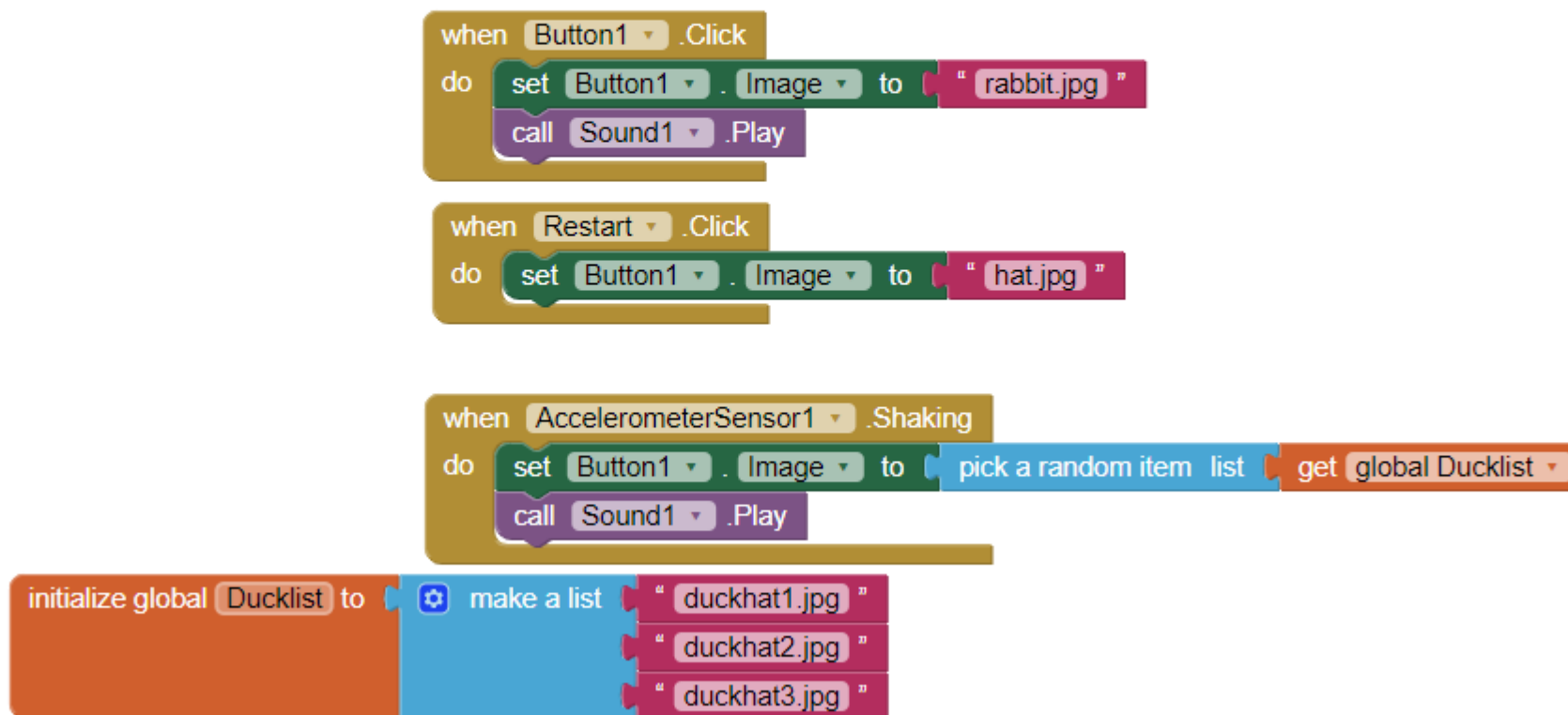
▶ טענו אפליקציה MAGIC

▶ אפליקציה עובדת עם תמונות שהכנו מראש



פתיחת אפליקציה קיימת - MAGIC

- ▶ בכל לחיצה על הכפתור , האפליקציה מנגנת צליל ומציגה תמונה
- ▶ בנוסף בהזזת טלפון האפליקציה מציגה שוב תמונה אקראית



פיתוח אפליקציות

שיעור מספר 2

שיעור 2 - בסיס APP INVERTOR , פקדים וכמובן הרבה אפליקציות

1. חזרה
2. עוד קצת על APP INVERTOR - תאוריה ורכיבים
3. ארכיטקטורת התוכנה
4. רכיבי ממשק המשתמש
5. כפתור ותכונותיו.
6. כפתור - הסבר גודל
7. כפתור - אירועים
8. תיבת טקסט - הכל
9. מעבר על אפליקציה הקיימת

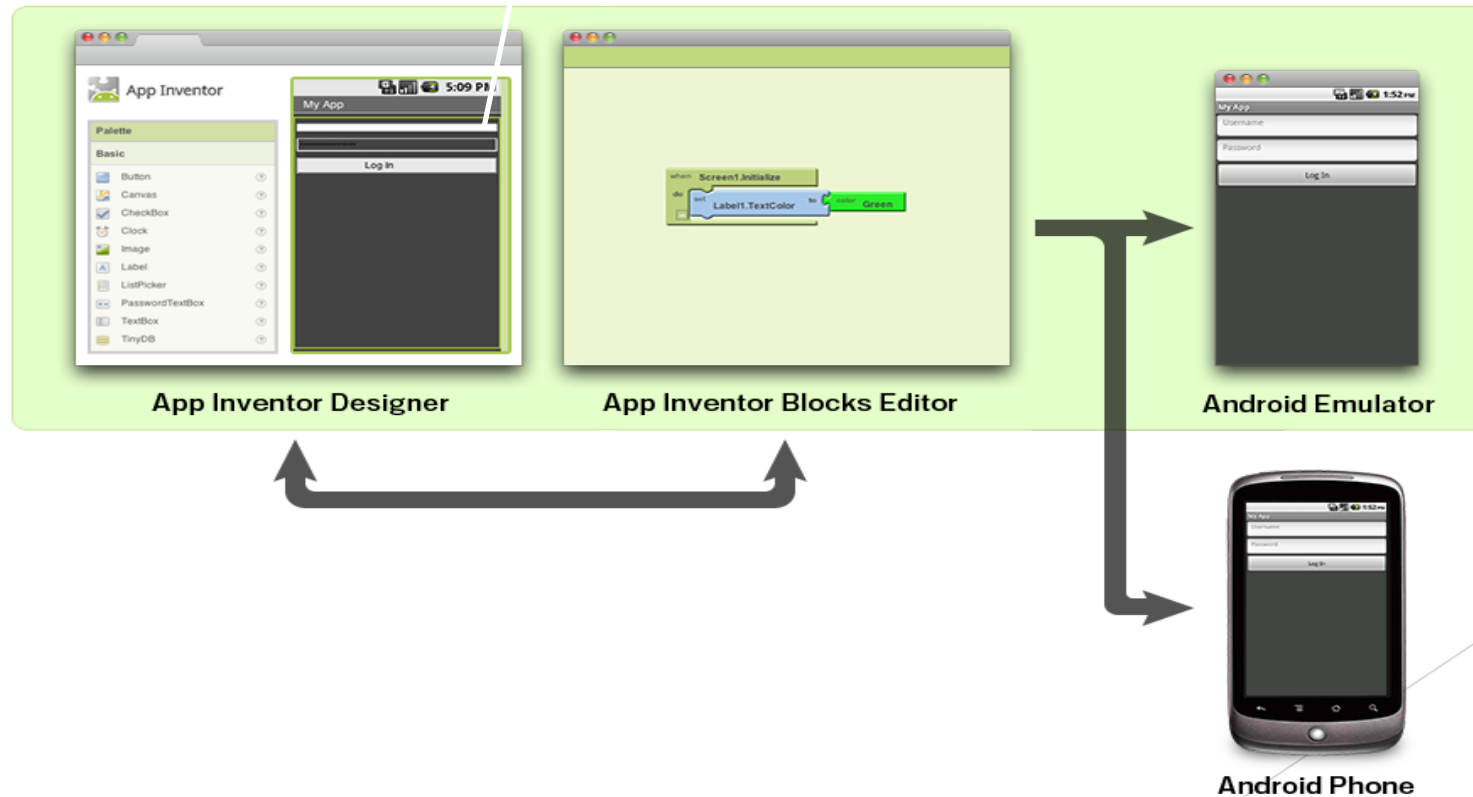
שרתי ה APP INVERTOR מאחסנים את העבודות שלך
וגם עוזרים לך לעקוב אחר הפרויקטים שלך.



Google App Inventor Servers



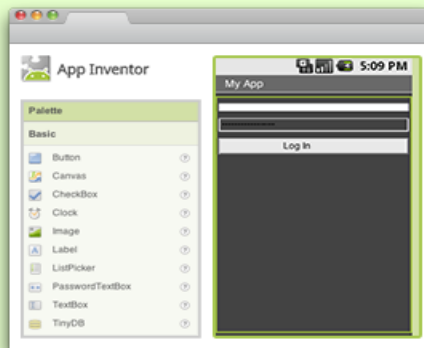
• APP INVERTOR מאפשר לך לפתח יישומים
עבור טלפונים אנדרואיד באמצעות דפדפן אינטרנט
או טלפון מחובר או אמולטור.



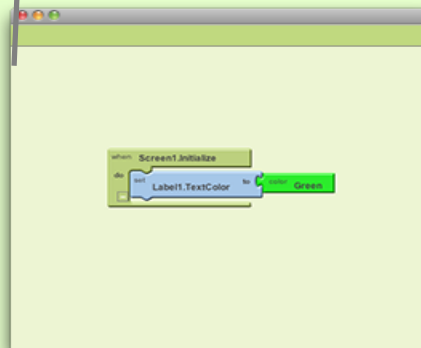
ממשיכים להכיר APP INVERTOR



Google App Inventor Servers



App Inventor Designer



App Inventor Blocks Editor



Android Emulator



Android Phone

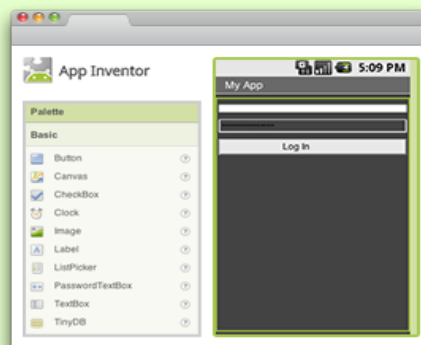
אתה בונה אפליקציות על ידי עבודה:
APP INVERTOR, שבו אתה בוחר את הרכיבים עבור
האפליקציה שלך.
APP BLOCK DESIGNER, שבו אתה יכול להרכיב
בלוקים התוכנית לציין כיצד הרכיבים צריכים להתנהג.



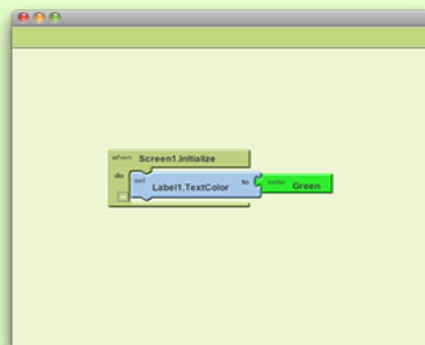
Google App Inventor Servers



- האפליקציה שלך מופיעה בטלפון צעד אחר צעד
- אתה מוסיף חתיכות של הבלוקים ואתה יכול לבדוק את התהליך שאתה בונה.
- כשתסיים, תוכל לארוז את האפליקציה שלך וליצור יישום עצמאי להתקנה.
- אם אין לך טלפון אנדרואיד, אתה יכול לבנות את היישומים שלך באמצעות אמולטור אנדרואיד, תוכנה הפועלת במחשב ומתנהג בדיוק כמו הטלפון.



App Inventor Designer



App Inventor Blocks Editor



Android Emulator



Android Phone

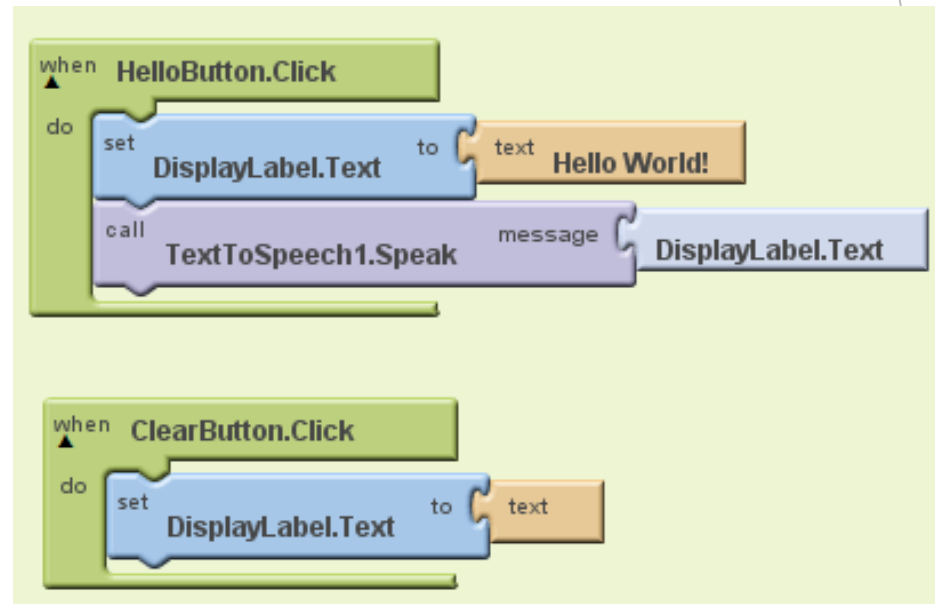


ההשוואה בין Java ל ApplInventor

Java Code

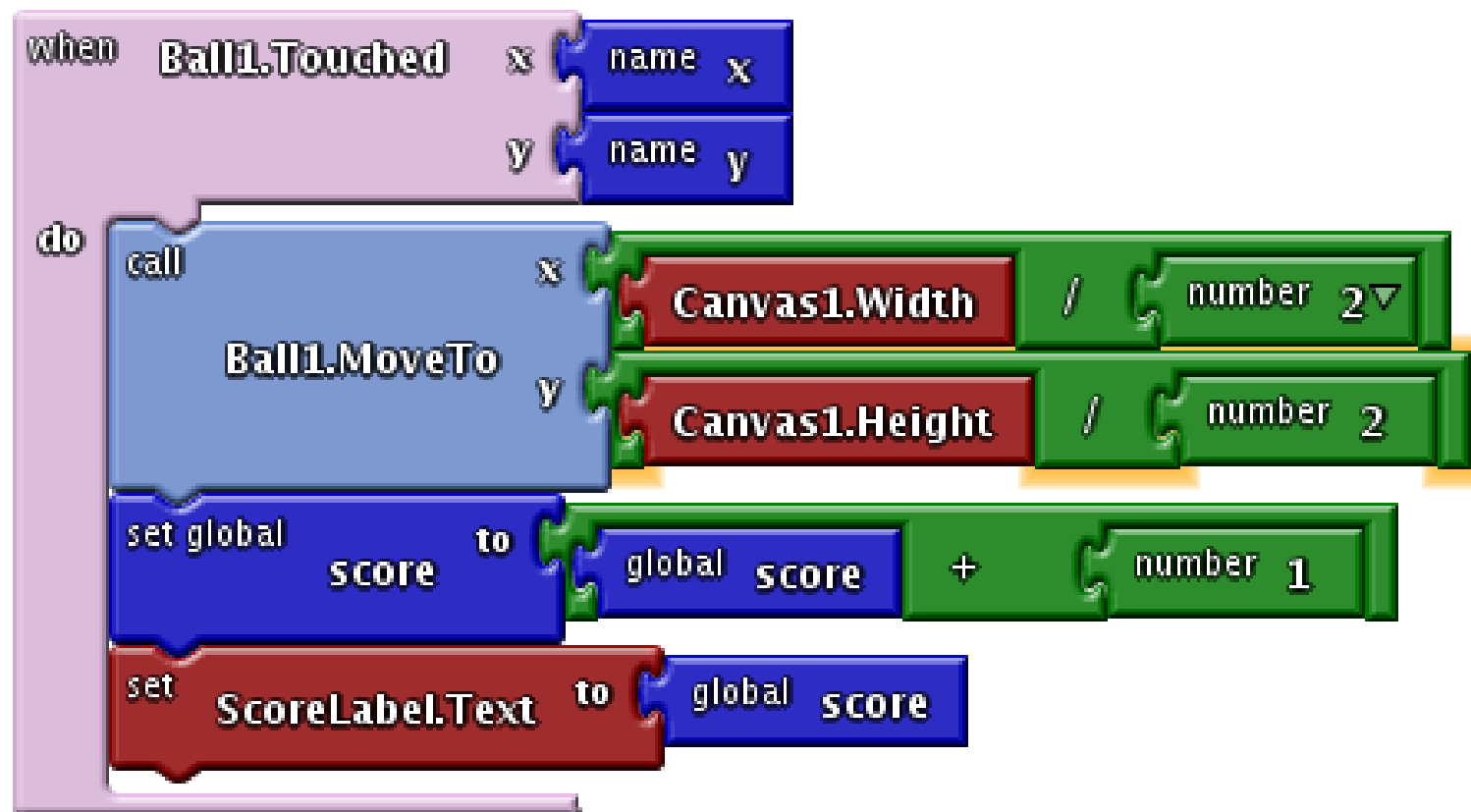
```
public class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

ApplInventor

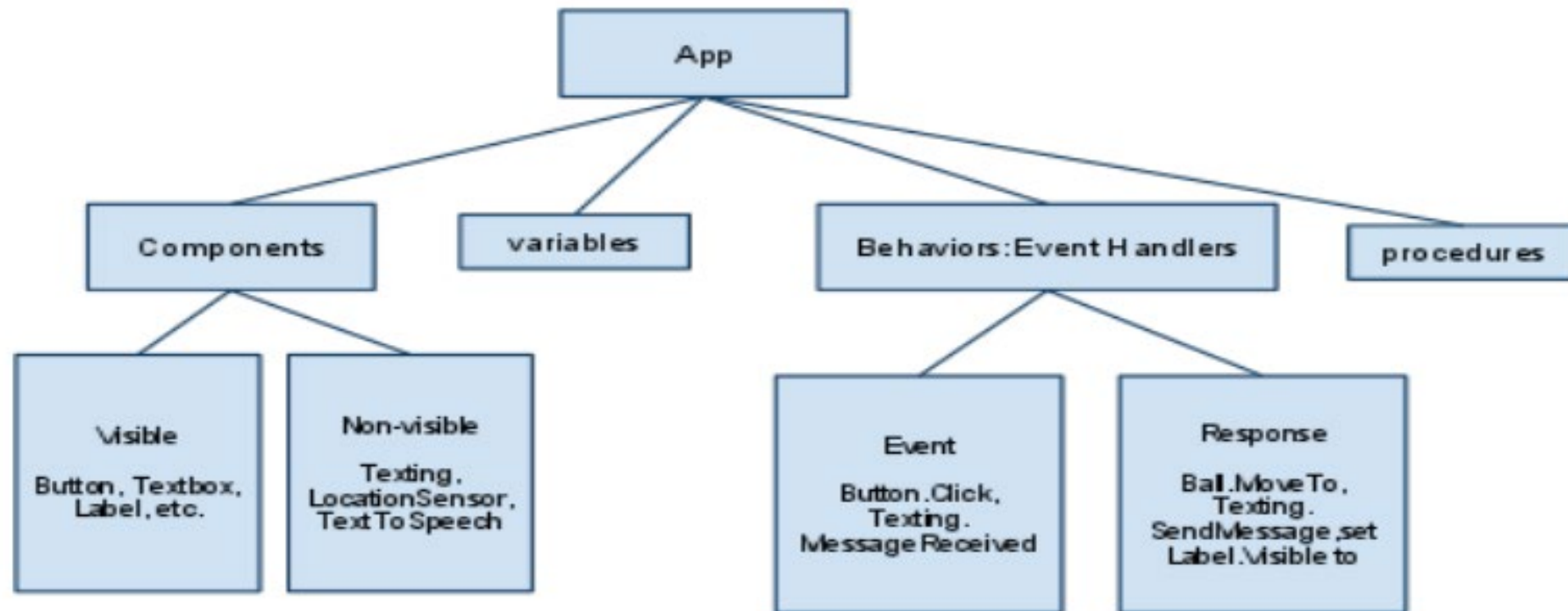


בלוקים

נחשו מה עושים הבלוקים הללו ?



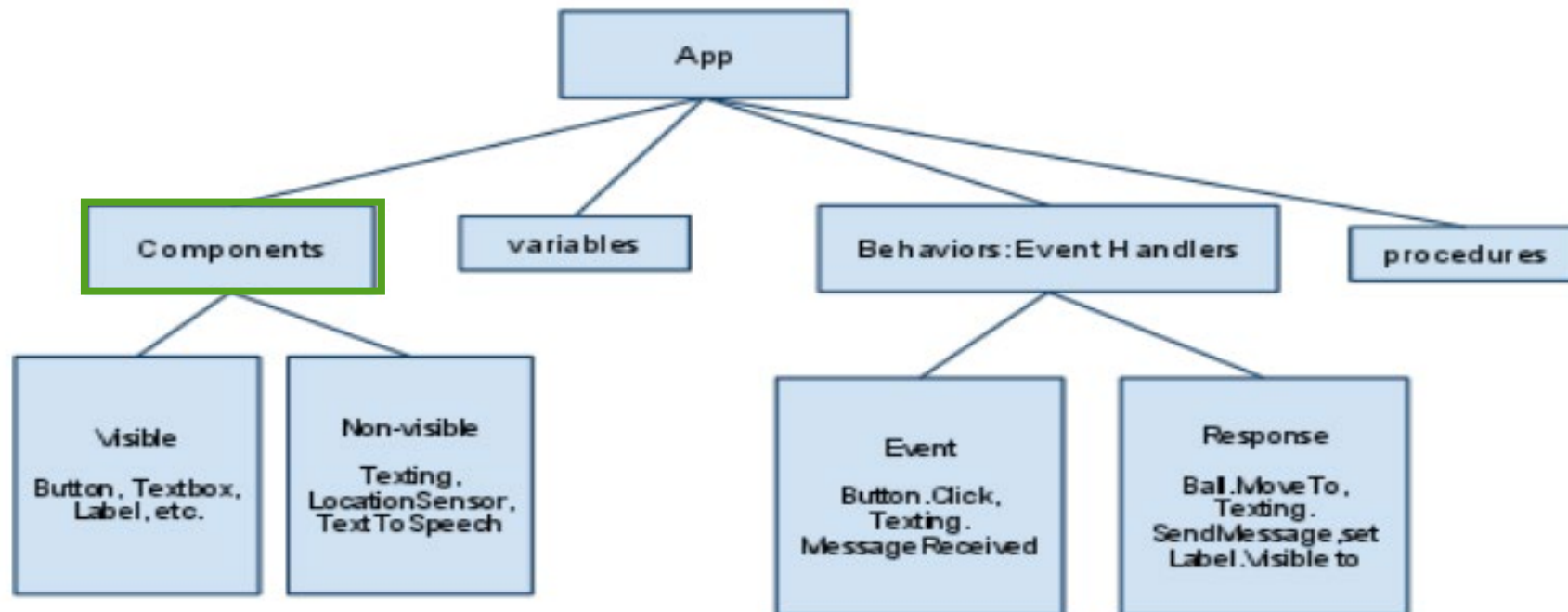
ארכיטקטורת התוכנית - ממה מורכבת האפליקציה??



רכיבים - COMPONENTS - הקומפוננטות

רכיבים הם אובייקטים או רכיבים המשמשים ליצירת יישום.

רכיבים יכולים להיות ויזואליים כמו כפתורים, תוויות או לא ויזואליים כמו למשל חיישנים



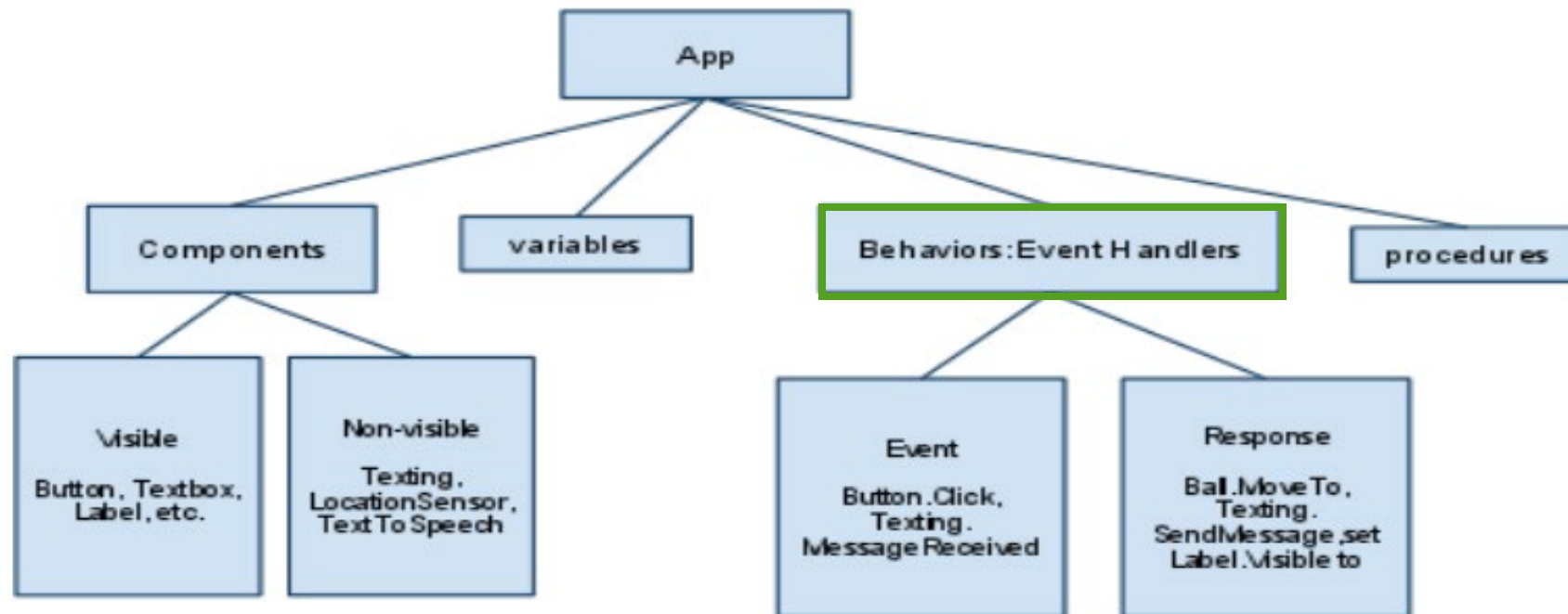
ארכיטקטורת התוכנה

- ▶ בתכנות מחשבים, תכנות מונחה-אירועים או תכנות מבוסס-אירועים
- ▶ תכנות שבו זרימת התוכנית נקבעת על-ידי האירועים-לדוגמא - פלטי חיישן או פעולות משתמש (לחיצות עכבר, הקשות מפתח) או הודעות מתוכניות אחרות
- ▶ למשל - אם משתמש לחץ על הכפתור תעשה....
- ▶ אם האפליקציה נפתחה, תעשה ...
- ▶ כל 20 שניות תעשה....
- ▶ אם התקבלה שיחת טלפון - תעשה

Event Type	Example
User-initiated event	when the user clicks button1 do...
Initialization event	when the app launches do...
Timer events	when 20 milliseconds passes do...
External events	when the phone receives a text do...

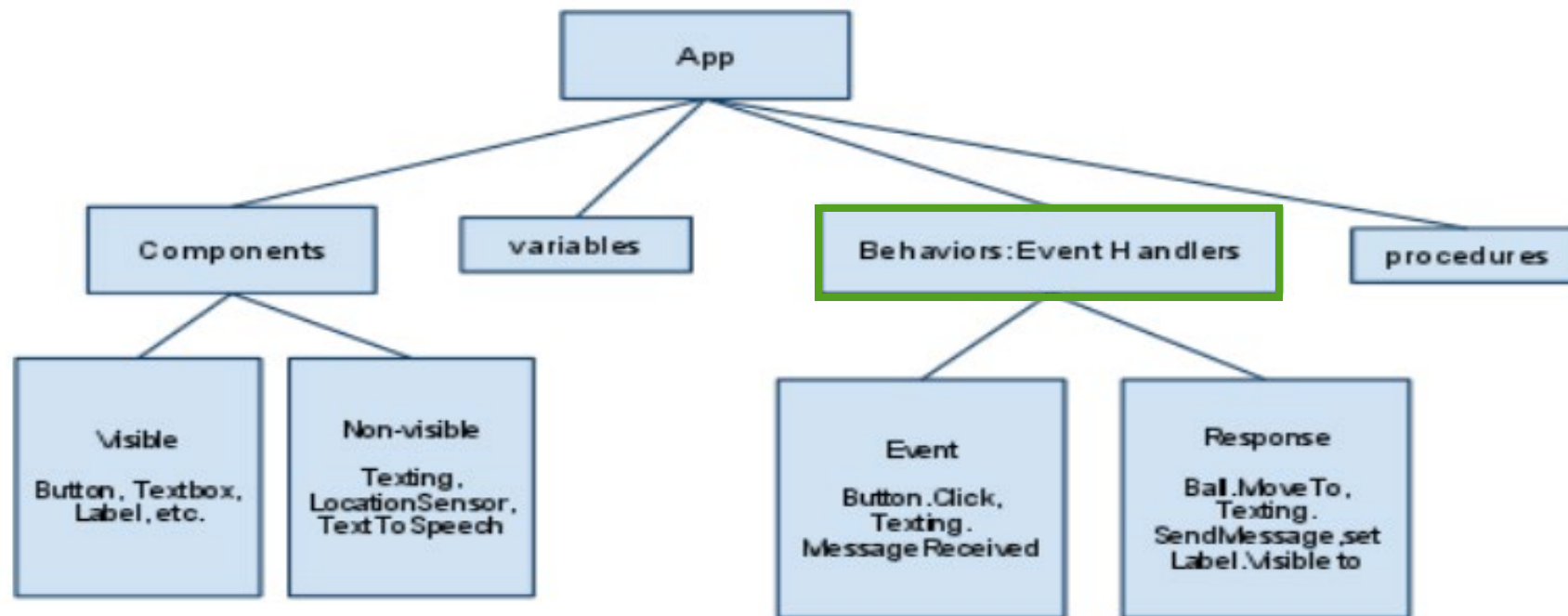
התנהגות - טיפול באירוע - Event handler

► הפונקציות שבוצעו בתגובה לאירוע. כשאירוע מתרחש, המטפל באירועים המתאים מופעל.



Behaviors - התנהגויות

▶ התנהגות - מגדירה כיצד האפליקציה צריכה להגיב על האירועים, שהמשתמש יזם (למשל, לחצן לחיצה) או אירועים חיצוניים כמו למשל הודעת טקסט SMS המגיע לטלפון



Why App Inventor Works

מדוע קל לעבוד עם APP INVENTOR ??

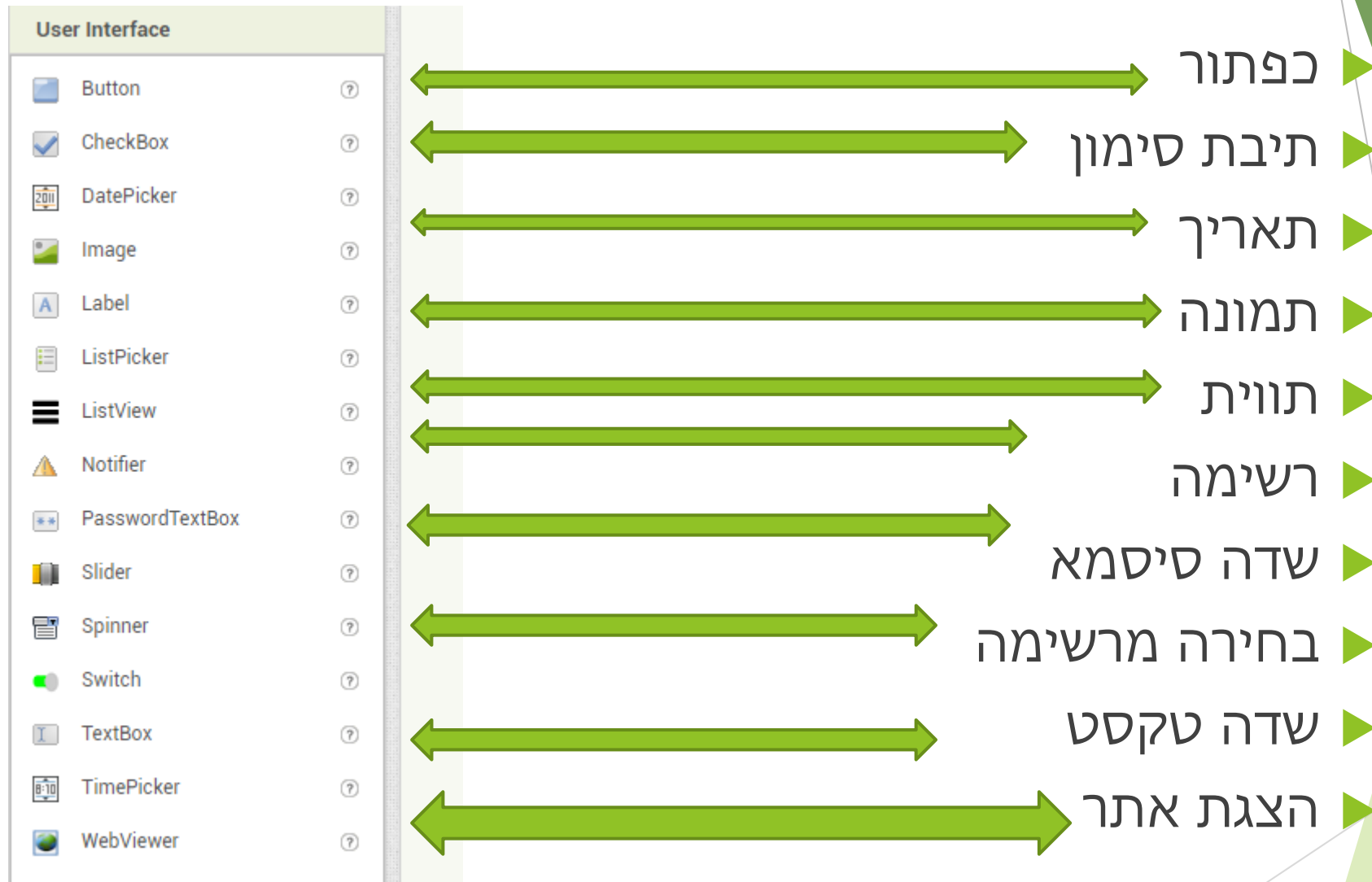
- ▶ אין הקלדה של קוד, אין שגיאות תחביר.
- ▶ אירועים ברמה הראשונה
- ▶ כמו להרכיב פאזל (רק להתאים כמה חתיכות)
- ▶ התכנות הוא ברמה פשוטה - צוות של גוגל הכניס בו הרבה עבודה
- ▶ מאוד ברור ו פחות מופשט

אבל החיים לא פשוטים....

- ▶ לא ניתן לבנות את הכל
 - ▶ user interface - או ממשק משתמש
 - ▶ לא בכל הטלפונים ניתן להפעיל כל האפשרויות
- ▶ תכנות זה עדיין עבודה קשה!

תכנות - משמעת קפדנית מבחינה אינטלקטואלית. שדורשת
הרבה תרגול!!

רכיבי ממשק משתמש בAPP INVERTOR



כפתור - BUTTON

- ▶ לחצן עם היכולת לזהות לחיצות - בכל לחיצה אפשר לבצע פעולה מסוימת
- ▶ ניתן לשנות היבטים הרבים של המראה שלו - צבע, גודל, רוחב וגובה.....
- ▶ ניתן לקבוע האם הוא זמין ללחיצה או לא
- ▶ לחצת - < אירוע
- ▶ עזבת - < אירוע
- ▶ עברת מעליו - < אירוע



תכונות הכפתור

נעבור ונסביר את
התכונות של הכפתור

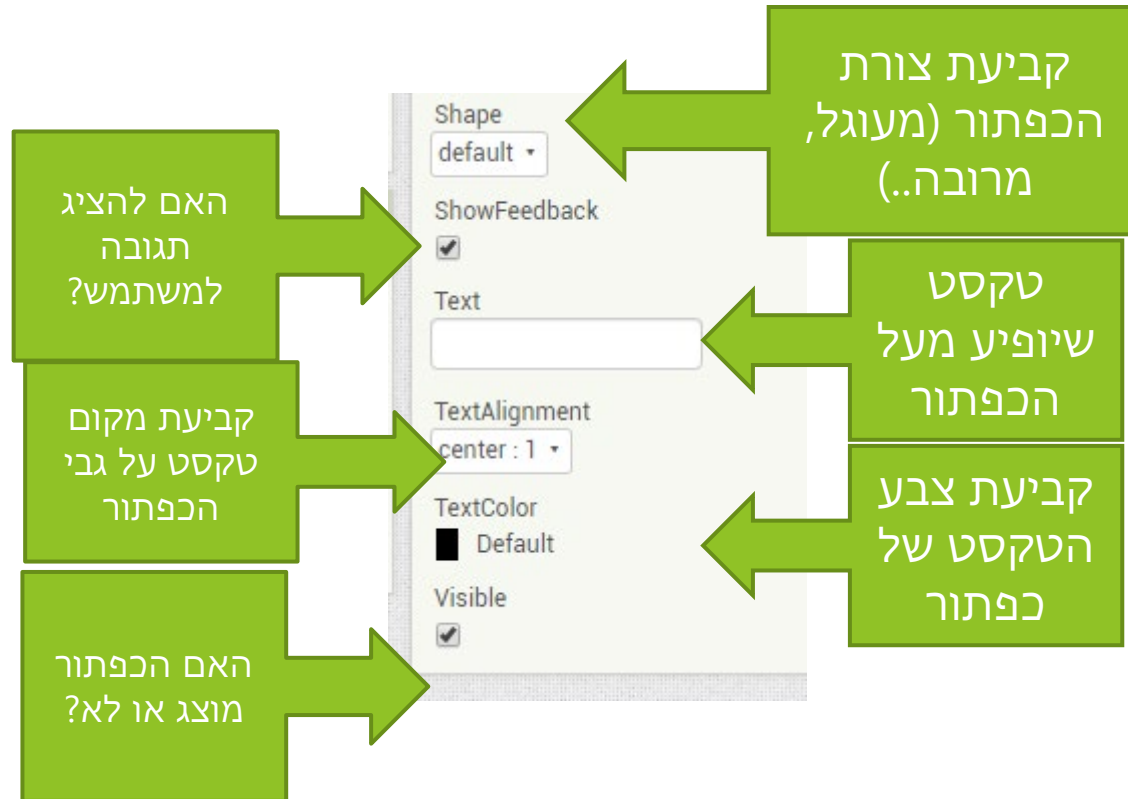
The image shows a software interface with a 'Components' panel on the left and a 'Properties' panel on the right. The 'Components' panel shows a hierarchy: Screen1 > Button1. The 'Properties' panel lists various attributes for Button1. Green arrows point from descriptive Hebrew text boxes to specific properties:

- קביעת צבע רקע של הכפתור** (Setting the background color of the button) points to **BackgroundColor** (Default).
- קביעת דגש לאותיות בכפתור** (Setting emphasis for letters on the button) points to **FontBold** (unchecked).
- קביעת גודל הטקסט** (Setting text size) points to **FontSize** (14.0).
- קביעת גובה הכפתור** (Setting button height) points to **Height** (Automatic...).
- קביעת רוחב הכפתור** (Setting button width) points to **Width** (Automatic...).
- קביעת נטייה לטקסט** (Setting text slant) points to **FontItalic** (unchecked).
- קביעת פונט לכפתור** (Setting font for the button) points to **FontTypeface** (default).
- קביעת צורה של הכפתור** (Setting button shape) points to **Shape** (default).
- קביעת תמונה מעל הכפתור** (Setting image above the button) points to **Image** (hat.jpg...).
- קביעת מראה של הכפתור** (Setting button appearance) points to **ShowFeedback** (checked).
- קביעת טקסט של הכפתור** (Setting button text) points to **Text**.

At the bottom left, there is a 'Media' section showing 'TaDasound.mp3'.

תכונות הכפתור

המשך התכונות...

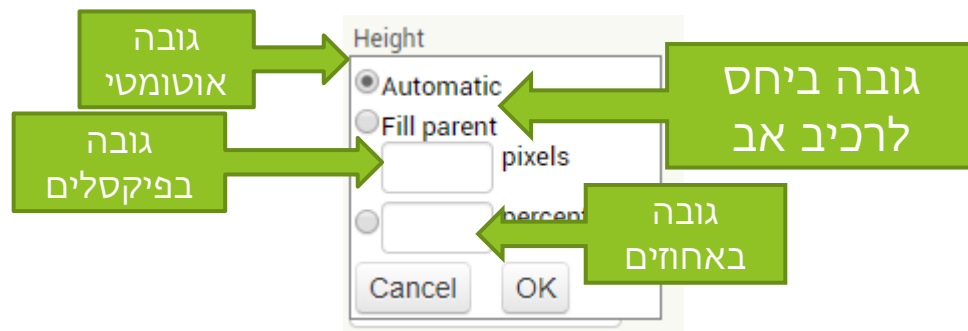


גובה ורוחב של הכפתור (ולא רק כפתור....)

- ▶ כפתור, כמו כל רכיב, נמצא תמיד מעל הרכיב אחר.
- ▶ הרכיב שהכפתור נמצא בתוכו - נקרא PARENT - אבא של הרכיב
- ▶ ניתן לקבוע לכפתור גובה ורוחב בכמה אפשרויות:
 - ▶ גודל ישיר בפיקסלים
 - ▶ גודל ישיר באחוזים
 - ▶ גודל ביחס לרכיב האבא שלו
 - ▶ וגודל אוטומטי

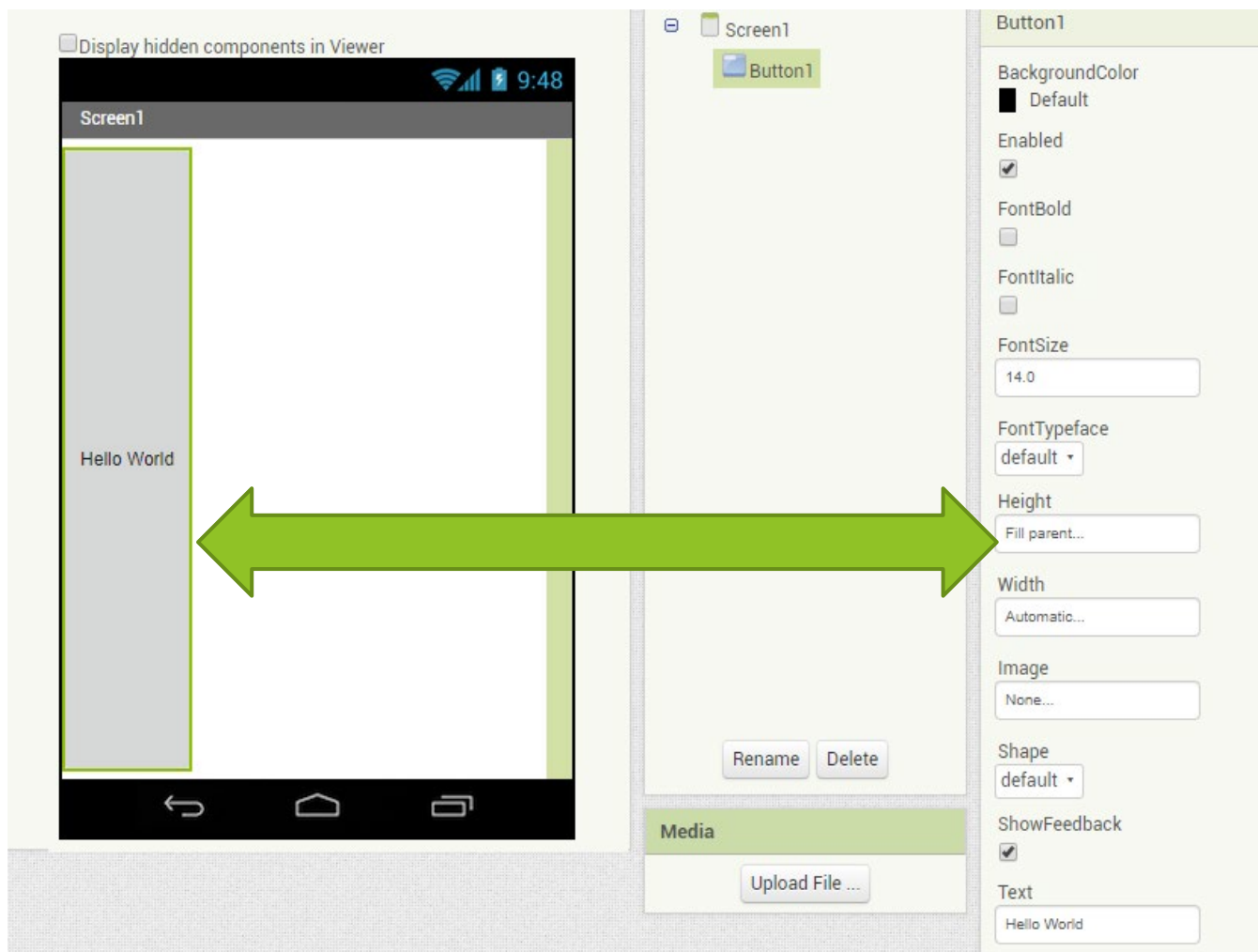
גובה ורוחב של הכפתור

נבחר את הכפתור ונצפה
אפשרויות גובה ורוחב הקיימות



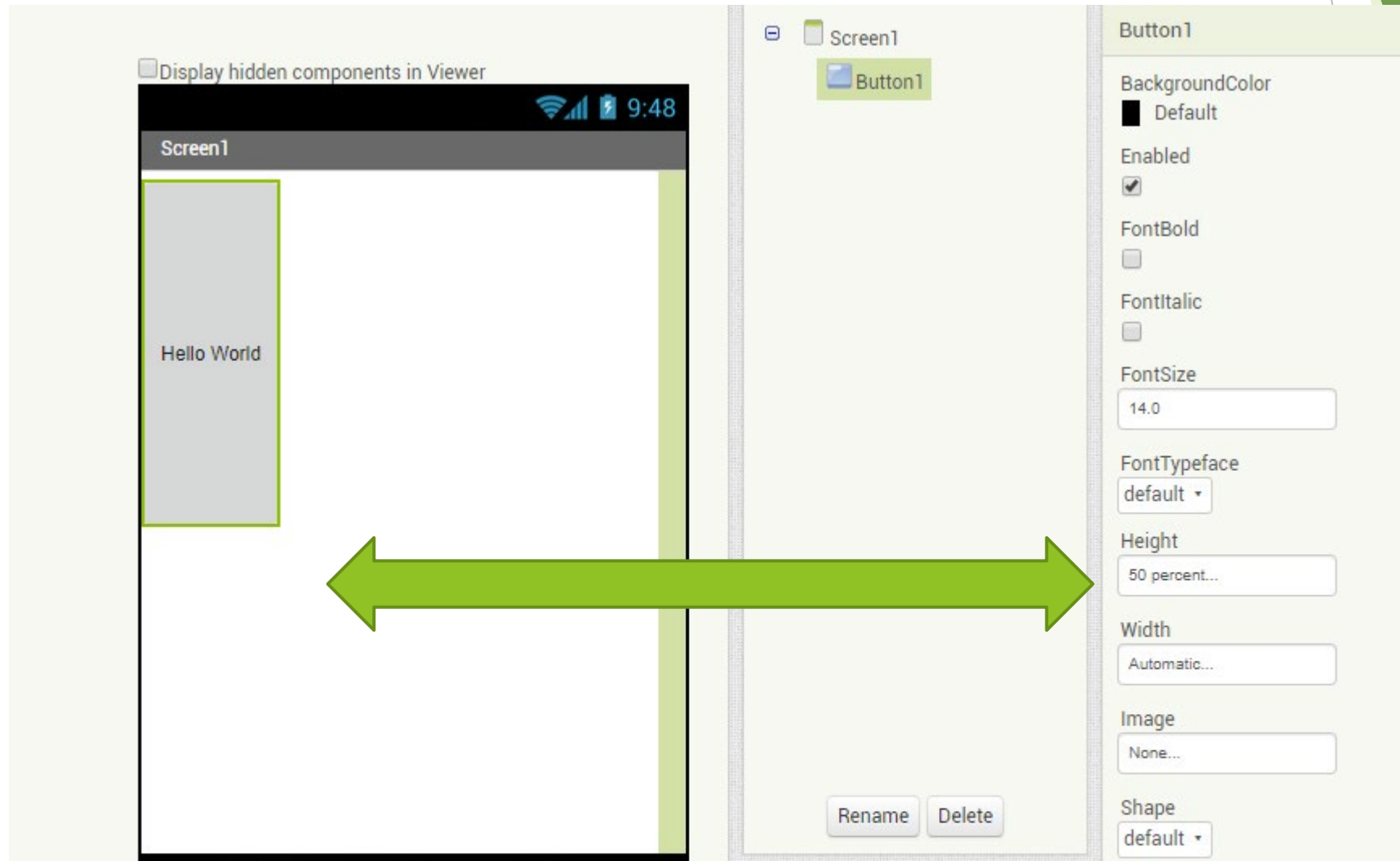
הסבר לגדלים

FILL PARENT - מלא את כל הרכיב האב... נניח כפתור נמצא על גבי המסך - במקרה כזה, הכפתור ימלא את כל המסך (בגובה בלבד!!)



הסבר לגדלים

50 אחוז... - הכוונה ל50 אחוז לפי גודל רכיב אב



תרגיל מספר 1

- ▶ הוסף על המסך כפתור הכחול וכתוב עליו את שמך.
- ▶ הוסף על המסך כפתור ברוחב 100 פיקסלים ובגובה 50 פיקסלים)
- ▶ הצג על המסך 3 כפתורים אחד מעל השני - הכפתורים מייצגים רמזור (ללא טקסט) . בצבעים - אדום, צהוב, ירוק
- ▶ הצג על המסך כפתור ומעליו יהיה כתוב טקסט הבא "**אני** **כפתור מרובה מוצג באדום, ITALIC ו BOLD**" - יש לגרום לטקסט להיות נכון
- ▶ הצג כפתור מוסתר (סודי שלא רואים אותו)
- ▶ הצג כפתור שמשתמש לא יוכל ללחוץ עליו

אירועים הכפתור BUTTON EVENTS

לאחר מעבר לBLOCK יש לבחור בBUTTON
לאחר בחירת BUTTON, יופיעו האירועים של
הכפתור

The screenshot displays the Scratch IDE interface. On the left, the 'Blocks' panel is visible, showing a list of built-in blocks categorized by type: Control, Logic, Math, Text, Lists, Colors, Variables, Procedures, Screen1, and Button1. The 'Button1' block is highlighted with a red oval. Below the 'Blocks' panel are 'Rename' and 'Delete' buttons. The 'Media' panel at the bottom left includes an 'Upload File ...' button. The main 'Viewer' area on the right shows a sequence of code blocks for 'Button1'. The first six blocks are event triggers: 'when Button1 .Click', 'when Button1 .GotFocus', 'when Button1 .LongClick', 'when Button1 .LostFocus', 'when Button1 .TouchDown', and 'when Button1 .TouchUp'. Each event block is followed by a 'do' block. Below these are property blocks: 'Button1 . BackgroundColor', 'set Button1 . BackgroundColor to', 'Button1 . Enabled', 'set Button1 . Enabled to', 'Button1 . FontBold', 'set Button1 . FontBold to', 'Button1 . FontItalic', and 'set Button1 . FontItalic to'.

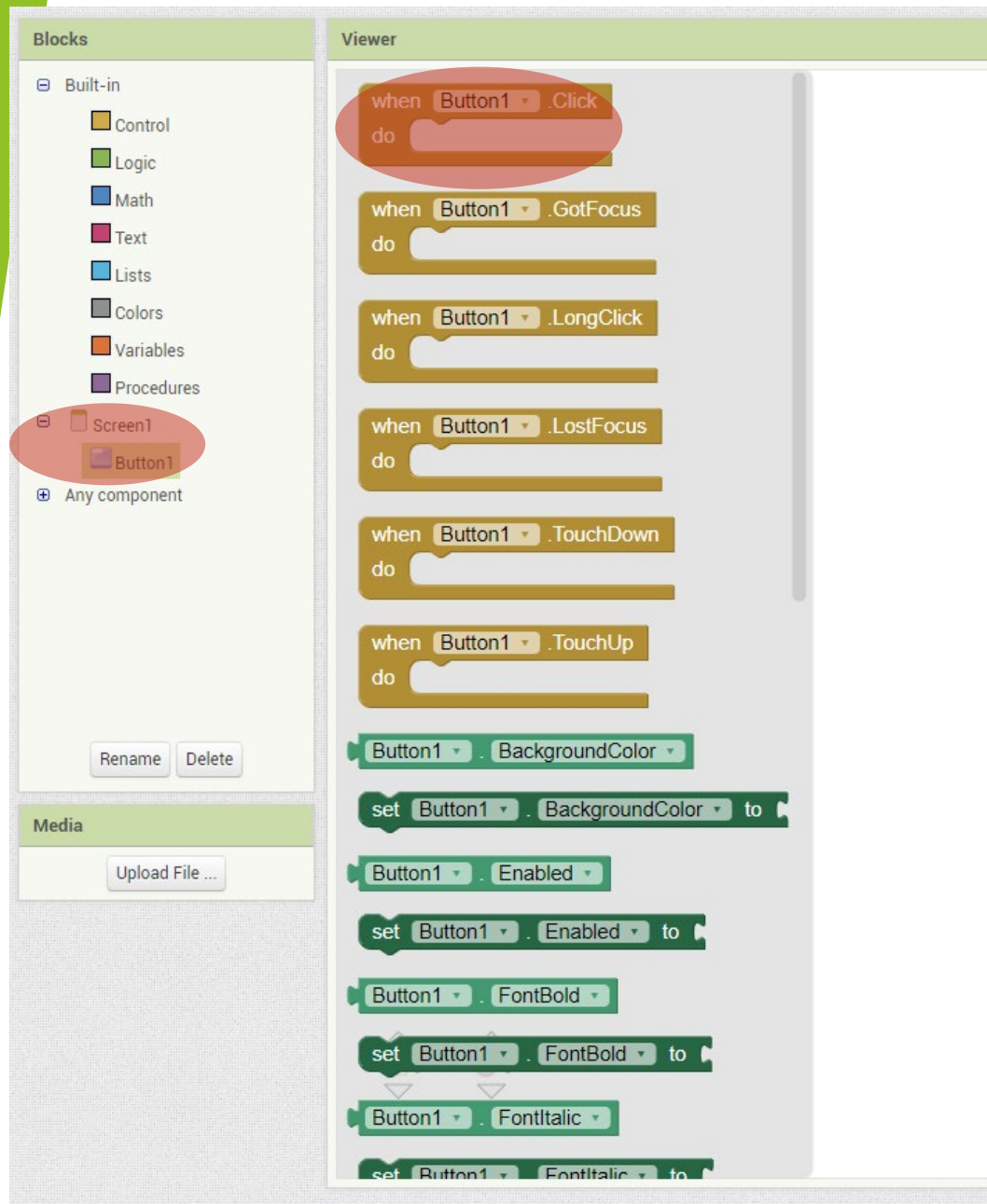
אירועים הכפתור BUTTON EVENTS

כאמור ניתן להפעיל אירוע בלחיצה על הכפתור.

על מנת להפעיל אירוע יש לפתוח את החוצץ השני - חוצץ הBLOCKS



אירועים הכפתור BUTTON EVENTS



- ▶ לאחר מעבר לBLOCK יש לבחור בBUTTON
- ▶ לאחר בחירת BUTTON, יופיעו האירועים של הכפתור
- ▶ האירועים הם בצבע **אפור**
- ▶ האירועים מתחילים עם המילה WHEN "כאשר"
- ▶ האירועים יסתיימו עם המילה DO "תעשה"
- ▶ ניתן לגרור אירוע מסוים למרכז המסך... למשל לגרור אירוע WHEN BUTTON CLICK

רשימת האירועים הכפתור BUTTON EVENTS

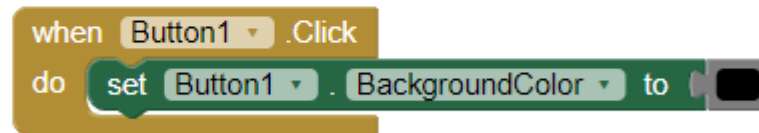
- ▶ `Click()` - המשתמש הקיש ושיחרר
- ▶ `GotFocus()` - מציין את הסמן שהועבר מעל ללחצן כך שכעת ניתן ללחוץ על
- ▶ `LongClick()` - המשתמש החזיק את הלחצן כלפי מטה.

רשימת האירועים הכפתור BUTTON EVENTS

- ▶ `LostFocus()` - מציין שהסמן התרחק מהלחצן. כך שכעת לא ניתן עוד ללחוץ עליו.
- ▶ `TouchDown()` - מציין שהלחצן נלחץ.
- ▶ `TouchUp()` - מציין שלחצן שוחרר.

מה ניתן לבצע בכל אירוע...

- ▶ בעת הלחיצה על הכפתור, ניתן לבצע פעולות שונות באפליקציה.
- ▶ למשל ניתן לעדכן ולשנות טקסט, לשנות את צבעי רקע וטקסט
- ▶ על מנת להפעיל את הכפתור יש להעביר את הפעולה המבוקשת לתוך DO של הכפתור
- ▶ למשל אם רוצים לעדכן את צבע הכפתור לשחור



איך משנים ערך?

מחזיר ערך של
צבע הכפתור

משנה ערך של
צבע הכפתור



שינוי ערך מתבצע עם פעולות SET

לאחר בחלק של הBLOCKS לאחר

בחירת כפתור מופיעים בנוסף לאירועים גם תיבות בירוק

התיבות הללו מתחלקות ל 2 חלקים GET וSET

GET - מחזיר ערך

SET משה ערך

שינוי צבע הכפתור

מחזיר ערך של
צבע הכפתור

משנה ערך של
צבע הכפתור

- ▶ גוררים לתוך EVENT בDO את SET של צבע רקע
- ▶ לאחר מכן בוחרים לאיזה צבע לשנות את הכפתור

בחירת
COLORS

בוחרים צבע
וגוררים אותם
לתוך
SET

מעדכנים את
צבע הכפתור
לשחור

```
when Button1 .Click  
do set Button1 . BackgroundColor to [black]
```

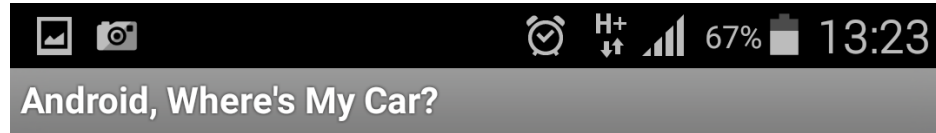

- ▶ צור אפליקציה אשר מעדכנת צבעים.
- ▶ אפליקציה מציגה כפתור A עם צבע אדום - < בלחיצה היא מעדכנת את בצע הכפתור לשחור
- ▶ אפליקציה מציגה כפתור B עם צבע שחור - < במעבר מעל הכפתור - < תשנה את הצבע שלו ללבן
- ▶ אפליקציה מציגה גם כפתור C - < בבלחיצה כפולה על הכפתור - < תשנו את הכיתוב של הכפתור לCCC

מעבר על דוגמא קיימת – אפליקציה לחיפוש מכונית
אחרי החנייה

<http://appinventor.mit.edu/explore/content/android-wheres-my-car.html>

אפליקציה איפה המכונית שלי

WhereIsMyCar App??



Your Current Location

Address: HaTa'asiya Street 40
Yehud
Israel

GPS: 32.0273 , 34.90184

Remember My Current Location

Remembered Place

Address: HaTa'asiya Street 15
Yehud
Israel

GPS: 32.02744 , 34.90036

Show Directions From Current to Remembered

ממשק משתמש - "Where is my car"

The screenshot displays the Android Studio IDE with three main panels: Viewer, Components, and Properties.

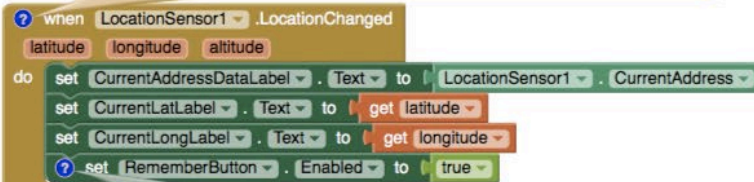
Viewer Panel: Shows a mobile app interface titled "Android, Where's My Car?". It features two sections: "Your Current Location" and "Remembered Place". Each section has an "Address: unknown" label, a "GPS: 0 , 0" label, and a button. The "Remembered Place" section also includes a "Show Directions From Current to Remembered" button. At the bottom, a "Non-visible components" section lists "LocationSensor1", "TinyDB1", and "ActivityStarter1".

Components Panel: Lists the UI components used in the app. It includes "Screen1" with sub-components like "CurrentHeaderLabel", "HorizontalArrangement2", "CurrentAddressLabel", "CurrentAddressDataLabel", "HorizontalArrangement1", "GPSLabel", "CurrentLatLabel", "CommaLabel", "CurrentLongLabel", "RememberButton", "RememberedHeaderLabel", "HorizontalArrangement3", "RememberedAddressLabel", "RememberedAddressDataLabel", "HorizontalArrangement4", "RememberedGPSLabel", "RememberedLatLabel", "RememberedCommaLabel", and "RememberedLongLabel". It also lists "DirectionsButton", "LocationSensor1", "TinyDB1", and "ActivityStarter1".

Properties Panel: Shows the properties for the selected "ActivityStarter" component. The "Action" is set to "android.intent.action.VIEW". The "ActivityPackage" is "com.google.android.apps.m". The "ActivityClass" is "com.google.android.maps.M".

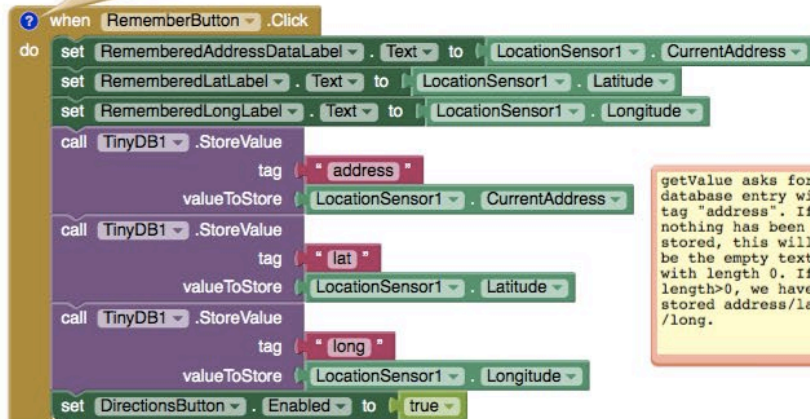
“Where is my car” - בלוקים

LocationChanged is triggered the first time the sensor gets a location reading and each time the location changes.



Once you have a reading, let the user "remember" it.

Put the current address in the "remember" labels of the UI and also remember this data in the database so it will be there when app is re-opened.



getValue asks for a database entry with tag "address". If nothing has been stored, this will be the empty text with length 0. If length>0, we have a stored address/lat/long.

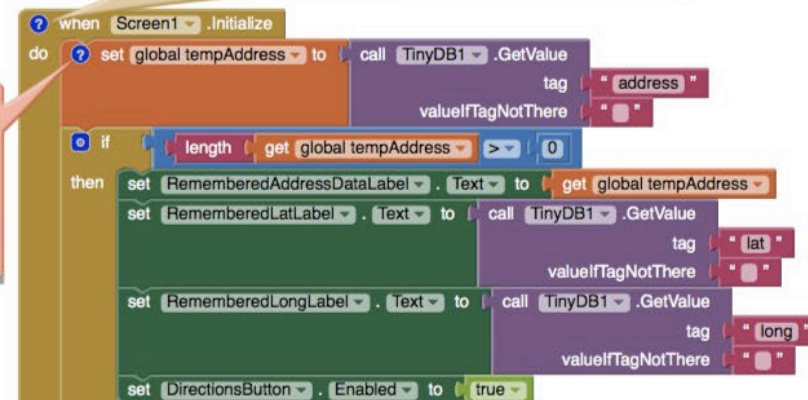
When the user asks for directions, build a maps URL with the current and remembered locations and call ActivityStarter to launch maps.



Join builds a text object from multiple parts. The result of this one is a URL that looks something like: <http://maps.google.com/maps?saddr=0.1,0.2&daddr=0.3,0.4>

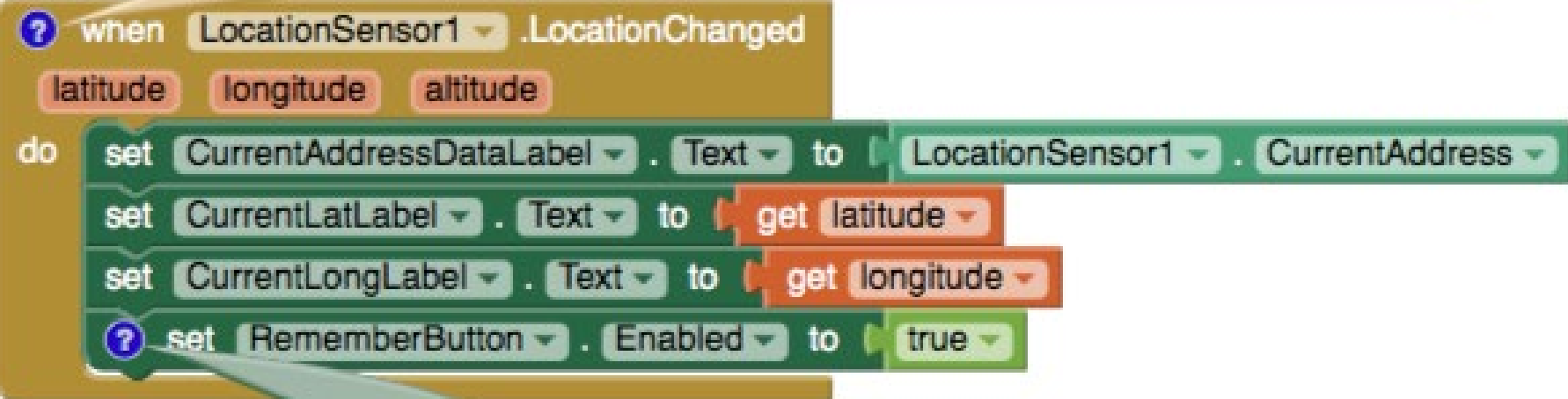
initialize global tempAddress to ""

When the app opens, see if a location has been remembered previously. If so, stick it in the remembered labels of the UI.



אירוע שינוי מיקום

LocationChanged is triggered the first time the sensor gets a location reading and each time the location changes.



Once you have a reading, let the user "remember" it.

אירוע - איתחול מסך

initialize global tempAddress to ""

When the app opens, see if a location has been remembered previously. If so, stick it in the remembered labels of the UI.

when Screen1.Initialize do

set global tempAddress to call TinyDB1.GetValue tag "address" valueIfTagNotThere ""

if length get global tempAddress > 0 then

set RememberedAddressDataLabel.Text to get global tempAddress

set RememberedLatLabel.Text to call TinyDB1.GetValue tag "lat" valueIfTagNotThere ""

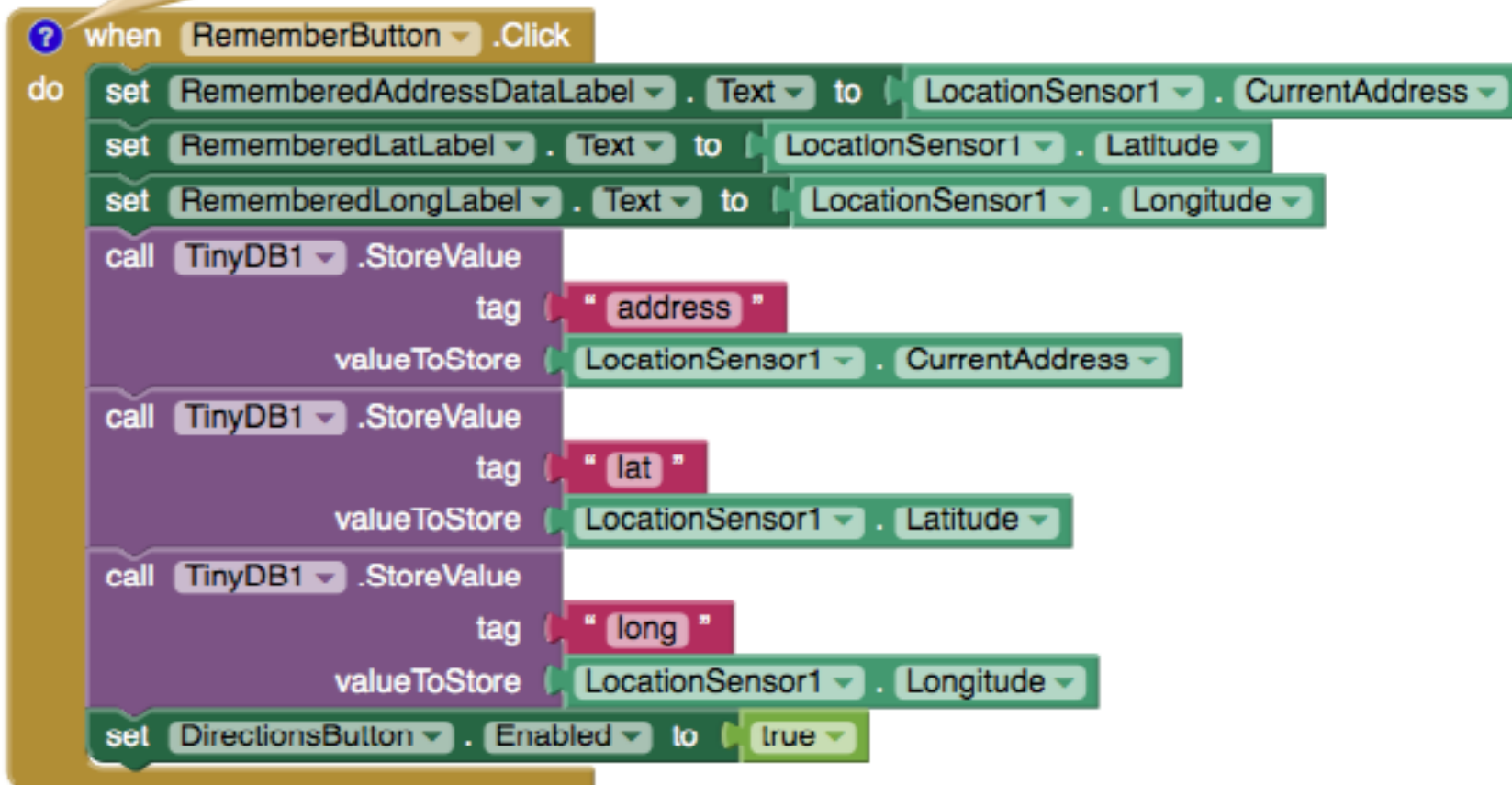
set RememberedLongLabel.Text to call TinyDB1.GetValue tag "long" valueIfTagNotThere ""

set DirectionsButton.Enabled to true

getValue asks for a database entry with tag "address". If nothing has been stored, this will be the empty text with length 0. If length>0, we have a stored address/lat/long.

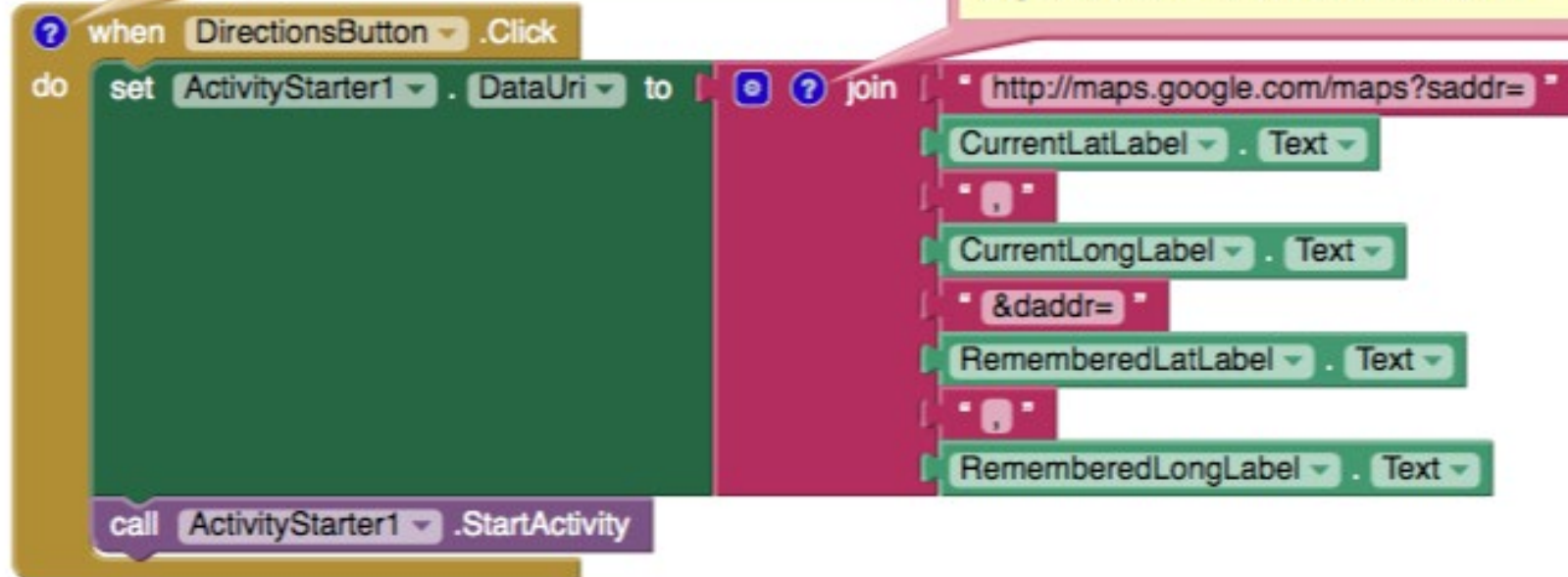
כפתור - לזכור את המיקום של המכונית

Put the current address in the "remember" labels of the UI and also remember this data in the database so it will be there when app is re-opened.



כפתור ניווט נלחץ

When the user asks for directions, build a maps URL with the current and remembered locations and call ActivityStarter to launch maps.



Join builds a text object from multiple parts. The result of this one is a URL that looks something like: `http://maps.google.com/maps?saddr=0.1,0.2&daddr=0.3,0.4`

[Clip for APP Inventor 1](#)

פיתוח אפליקציות

שיעור מספר 3

שיעור 3 - בסיס APP INVERTOR , תיבת טקסט , המון רכיבים ותרגול

- .1 חזרה
- .2 TEXTBOX
- .3 LAYOUT
- .4 DATEPICKER
- .5 SLIDER
- .6 SPINNER

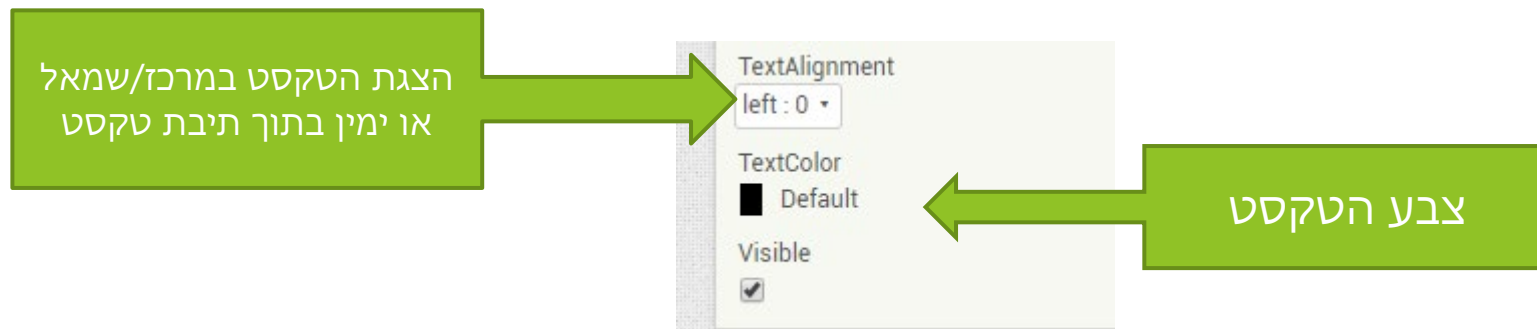
תרגיל מספר 1 - חזרה

- ▶ צור אפליקציה אשר בכל לחיצה על הכפתור מדליקה כפתור נוסף.
- ▶ מטרת המשחק - לגלות את כל הכפתורים על המסך
- ▶ בתחילת המשחק מוצג הכפתור הראשון, לאחר הלחיצה על הכפתור מוצגים 2 כפתורים הבאים וכך הלאה.
- ▶ מעל כל כפתור מופיע ערך נכון הכפתור - 1, 2, 3, 4, 5 וכך הלאה לפי הסדר
- ▶ או מופיע הערך הלא נכון 1,3,4,5,6,23,23,32
- ▶ במידה ונלחץ הכפתור הלא נכון, האפליקציה תשנה את כל הכפתורים לאדום

- ▶ לרב הרכיבים ישנם תכונות זהות כגון גובה, רוחב, צבע רקע, פונט ועוד
- ▶ לרב הרכיבים ישנם גם אירועים זהים כגון ONCLICK, ONDOUBLECLICK ועוד.
- ▶ יחד עם זאת, לכל רכיב יש גם תכונות ייחודיות משלו...
- ▶ מעתה נדבר רק על יכולות הייחודיות הללו.

- ▶ תפקידו של TEXTBOX באפליקציה הינו ביצוע קלט מהמשתמש
- ▶ ניתן לבצע קלט מספרים ואותיות
- ▶ ניתן לקבוע גודל הטקסט
- ▶ ניתן לקבוע האם הטקסט יהיה רק המספרים או רק האותיות?

תיבת טקסט - TEXTBOX



- ▶ GotFocus()-נקרא כאשר התיבה מקבלת פוקוס
- ▶ LostFocus()-נקרא כאשר התיבה מאבדת פוקוס

► HideKeyboard()-

הסתרת מקלדת ל-Multiline

► RequestFocus() -

לתת פוקוס לשדה (להפעילו)

תיבת טקסט - TEXTBOX

The screenshot displays the Android Studio environment. On the left, the 'Viewer' pane shows a mobile app preview with a status bar at 9:48, a header 'Screen1', a text input field labeled 'טקסט', and a large grey area with 'Hello World'. The middle pane shows the 'Screen1' component tree with 'TextBox1' and 'Button1'. The right pane shows the 'TextBox1' properties:

- BackgroundColor: Default
- Enabled: ☒
- FontBold: ☐
- FontItalic: ☐
- FontSize: 14.0
- FontTypeface: default
- Height: Automatic...
- Width: Fill parent...
- Hint: Hint for TextBox1
- MultiLine: ☐
- NumbersOnly: ☐
- Text: טקסט

Four green callout boxes with arrows point to specific features:

- Box 1 (points to the 'טקסט' label in the preview): רמז - הטקסט הזה מופיע בתיבה כל עוד המשתמש לא התחיל להזין טקסט משלו
- Box 2 (points to the 'MultiLine' checkbox): ניתן למנוע מהמשתמש להזין תווים - ספרות בלבד
- Box 3 (points to the 'NumbersOnly' checkbox): ניתן לקלט טקסט במספר שורות
- Box 4 (points to the 'Text' field): תוכן התחלתי של השדה - שימו לב לרב

LAYOUT - מיקום רכיבים על גבי המסך

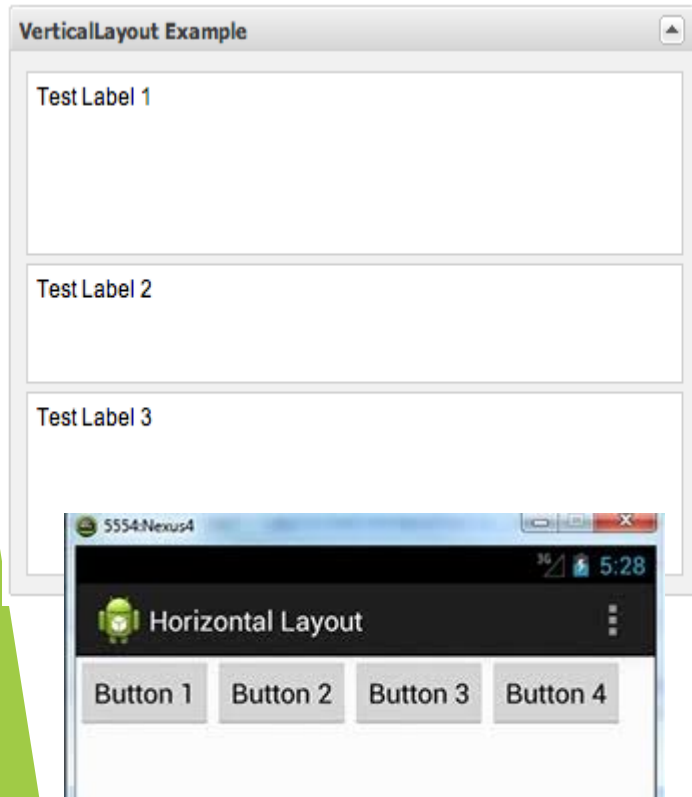
▶ על מנת לעזור לתוכניתן (לכם 😊) למקם את הרכיבים על גבי המסך קיימים רכיבים נוצרו הLAYOUTS.

▶ LAYOUTS - תפקידם לבצע סדר על גבי המסך.

▶ ב APP INVERTOR קיימים 2 סוגים LAYOUTS - אופקיים ואנכיים & VERTICAL HORIZONTAL

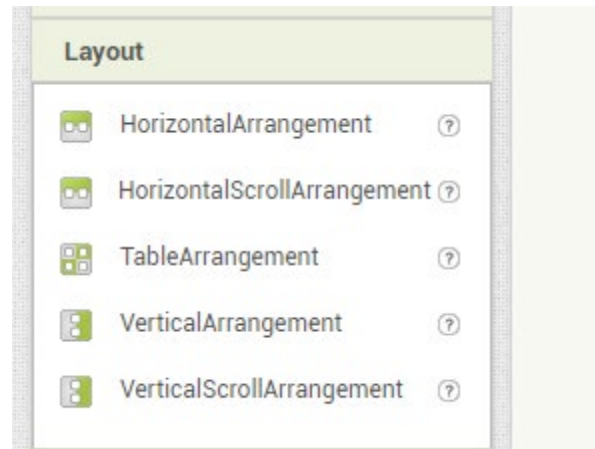
▶ בVERTICAL - כל רכיבים מסודרים אחד מתחת לשני

▶ בHORIZONTAL - כל הרכיבים מסודרים מימין לשמאל
(או משמאל לימין)



LAYOUT - מיקום הרכיבים על גבי המסך

על מנת לבחור LAYOUT יש לבחור באופציה LAYOUT בDESIGNER
ב APP INVERTOR קיימים LAYOUT הבאים:



- ▶ השתמש ברכיב סידור אנכי על מנת להציג קבוצה של רכיבים שהונחו משמאל לימין או מימין לשמאל

תכונת הגובה של רכיב האנכי

- ▶ אם המאפיין הגובה של HorizontalArrangement מוגדר כאוטומטי, הגובה של הסידור נקבע על-ידי הרכיב הגבוה ביותר.
- ▶ אם המאפיין גובה של HorizontalArrangement מוגדר כאוטומטי והוא מכיל רק רכיבים שמאפייני הגובה שלהם מוגדרים כמילוי הורה - FILL PARENT, הגובה הממשי של הסידור מחושב באמצעות הגבהים האוטומטיים של הרכיבים.
- ▶ אם המאפיין גובה של HorizontalArrangement מוגדר כאוטומטי והוא ריק, הגובה יהיה 100.

▶ השתמש ברכיב סידור אנכי כדי להציג קבוצה של רכיבים שהונחו מלמעלה למטה, מיושר לשמאל.

תכונות הגובה של רכיב אופקי:

▶ המאפיין הרוחב של `VerticalArrangement` מוגדר כאוטומטי, הרוחב בפועל של הסידור נקבע על-ידי הרכיב הרחב ביותר בסידור שהמאפיין `Width` שלו אינו מוגדר כמילוי הורה - `FILL PARENT`.

▶ אם המאפיין רוחב של `VerticalArrangement` מוגדר כאוטומטי והוא מכיל רק רכיבים שמאפייני הרוחב שלהם מוגדרים כמילוי הורה - `FILL PARENT`, הרוחב בפועל של הסידור מחושב באמצעות הרוחב האוטומטי של הרכיבים.

▶ אם המאפיין רוחב של `VerticalArrangement` מוגדר כ אוטומטי והוא ריק, הרוחב יהיה 100.

TABLE ARRANGMENT

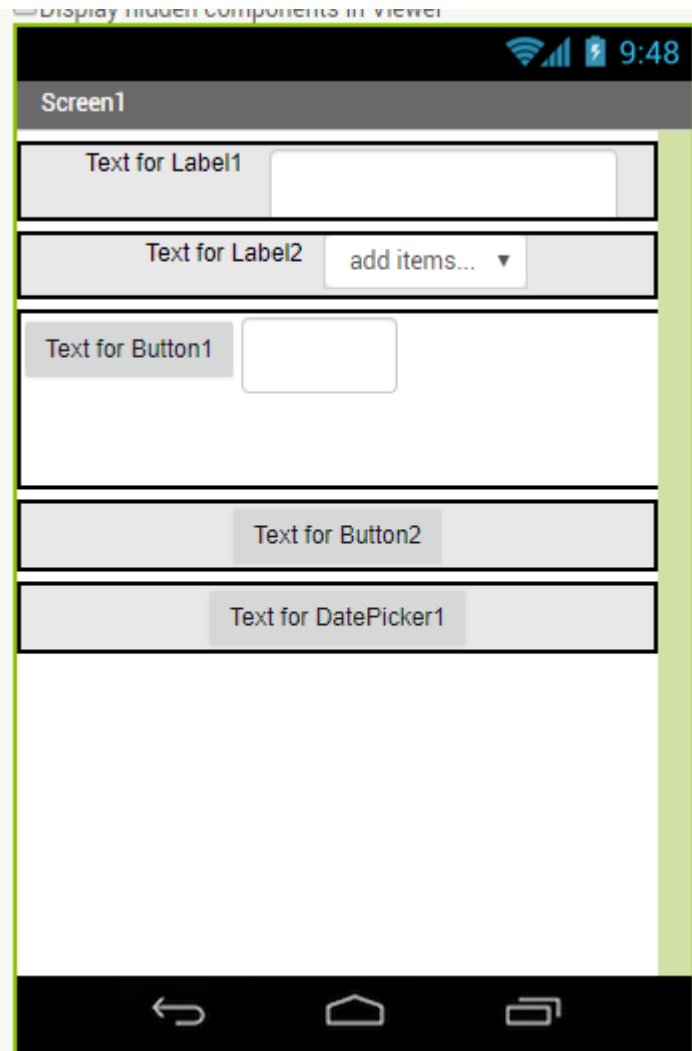
- ▶ השתמש ברכיב סידור טבלה כדי להציג קבוצת רכיבים בצורה טבלאית.
- ▶ רכיב זה הוא רכיב עיצוב שבו מניחים רכיבים שיש להציגם בצורה טבלאית.
- ▶ ב `TableArrangement` הרכיבים מסודרים ברשת של שורות ועמודות, כאשר לא יותר מרכיב אחד גלוי בכל תא. אם רכיבים מרובים תופסים את אותו תא, רק האחרון יהיה גלוי.

TABLE ARRANGMENT

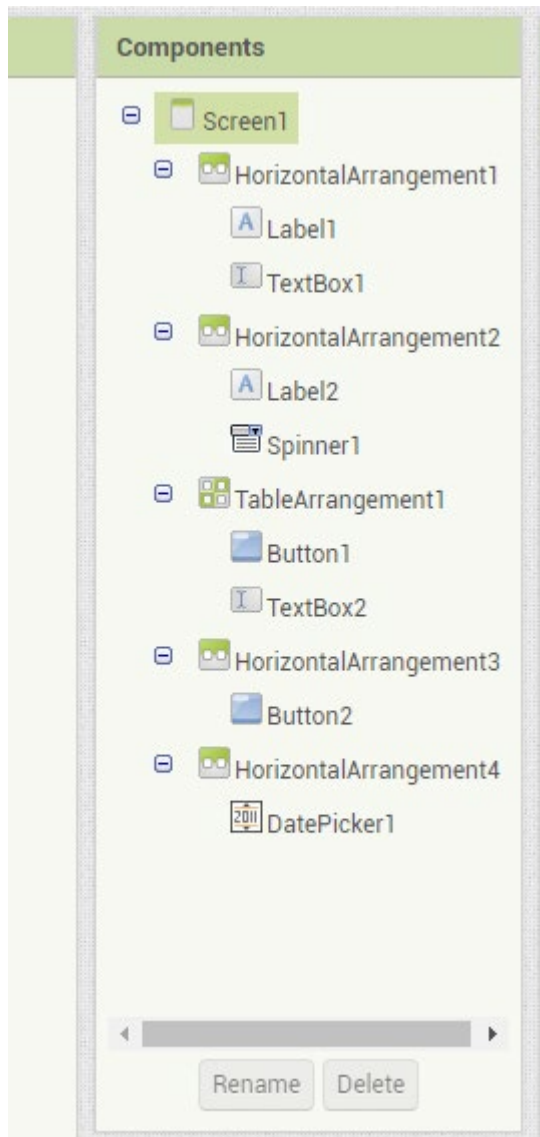
- ▶ בתוך כל שורה, הרכיבים מיושרים למרכז באופן אנכי.
- ▶ רוחב העמודה נקבע על-ידי הרכיב הרחב ביותר בעמודה זו. בעת חישוב רוחב עמודה, הרוחב האוטומטי משמש עבור רכיבים שהמאפיין Width שלהם מוגדר כמילוי הורה - FILL PARENT. עם זאת, כל רכיב ימלא תמיד את הרוחב המלא של העמודה שהיא תופסת.
- ▶ גובה השורה נקבע על-ידי הרכיב הגבוה ביותר באותה שורה שהמאפיין Height שלו אינו מוגדר כמילוי הורה - FILL PARENT. אם השורה מכילה רק רכיבים שמאפייני הגובה שלהם מוגדרים כמילוי הורה - FILL PARENT, גובה השורה מחושב באמצעות הגבהים האוטומטיים של הרכיבים.

תרגיל מספר 2 - מיקום רכיבים על גבי המסך

יש לבנות ולמקם רכיבים בצורה הבאה: ►



תרגיל מספר 2 - פתרון

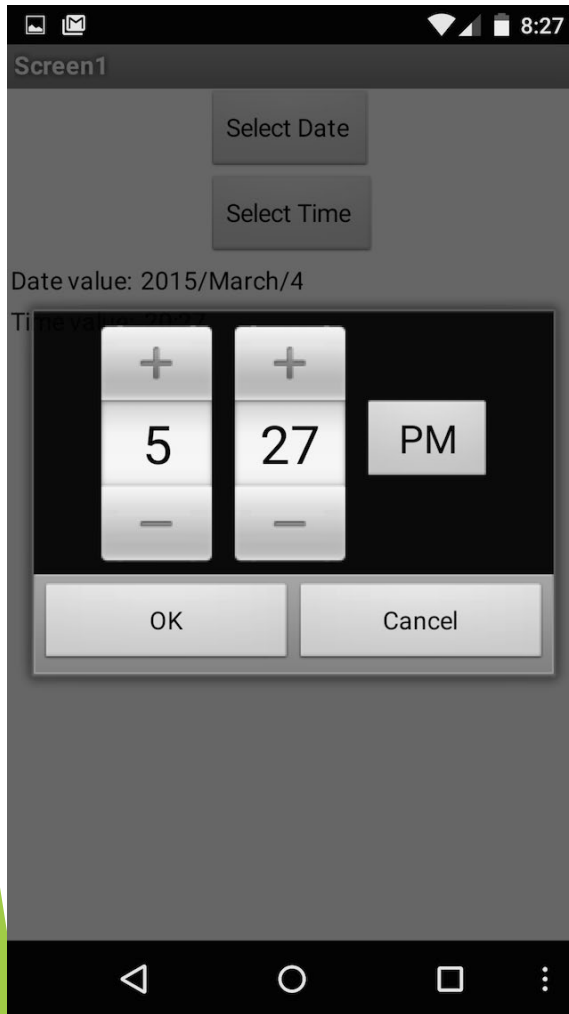


שימו לב - < על מנת למקם 2 רכיבים בשורה יצרנו 4 LAYOUT ים רוחביים.

בנוסף יצרנו LAYOUT אחד טבילאי.

בתוך LAYOUT ים הללו גררנו את הרכיבים שלנו.

תיבת בחירת תאריך DatePicker ותיבת בחירה זמן TIMEPICKER



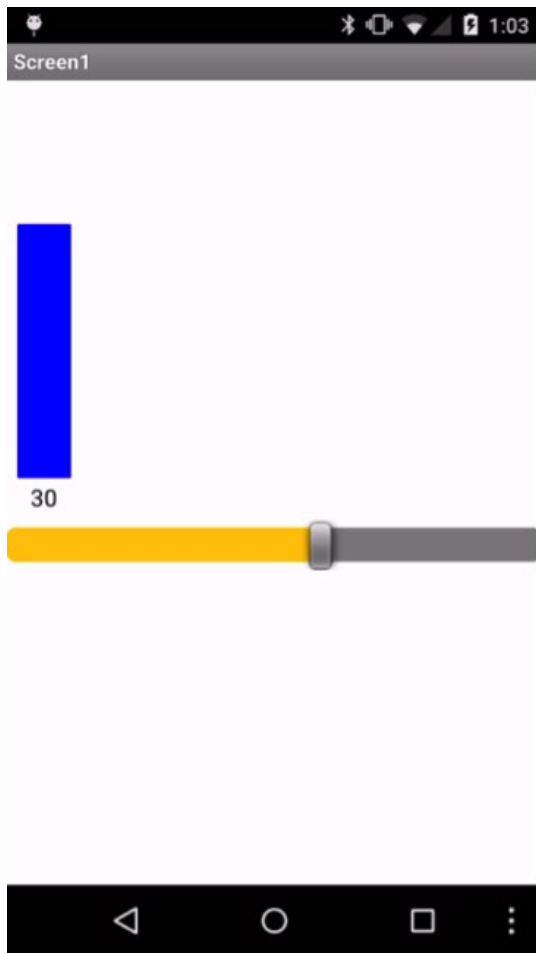
► לחצן שכאשר לחצת עליו, מציגה תיבת דו-שיח כדי לאפשר למשתמש לבחור את התאריך.

תכונות חשובות

- Day - היום בחודש שנבחר לאחרונה באמצעות בורר התאריכים.
- MONTH - מספר החודש שנבחר לאחרונה באמצעות בורר התאריכים.
- שימו לב: חודשים מתחילים ב-1 = ינואר, 12 = דצמבר.
- YEAR - השנה שנבחרה באמצעות בורר התאריכים

בר SLIDER

▶ סלידר - מאפשר בחירת ערך. ניתן לגרור אותו ימינה ושמאלה ולדעת איזה ערך בחר המשתמש



תכונות חשובות

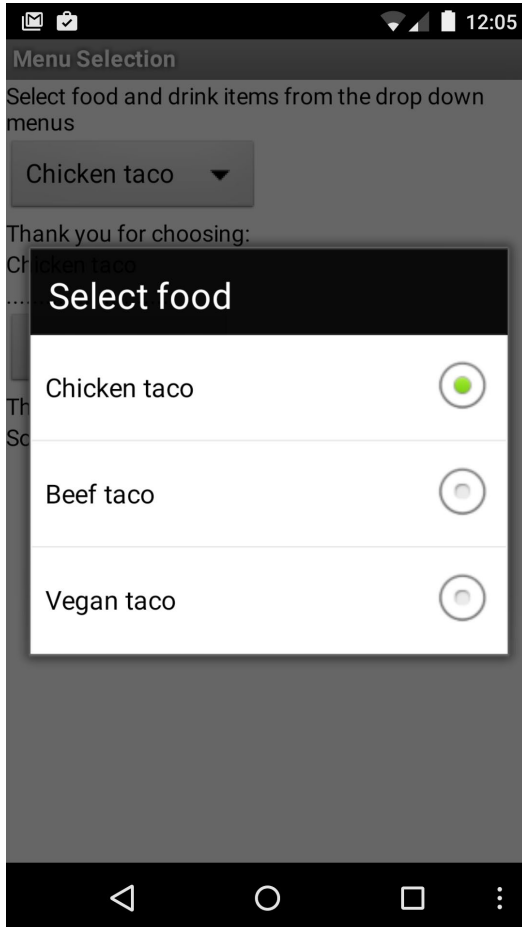
- ▶ בכל תזוזה של SLIDER, נורה אירוע POSITION CHANGED ומופיע בו מקום חדש.
- ▶ SLIDER מאפשר למשתמש דרך קלה לבחור ערך.
- ▶ SLIDER יש תכונות ערך מקסימלי וערך מינימלי וכמו כן יש ערך קפיצה בין מקסימלי למינימלי

SPINNER - תיבת בחירה

► תיבת הבחירה - SPINNER מאפשרת למשתמש לבחור ערך מתוך רשימת ערכים הנפתחת.

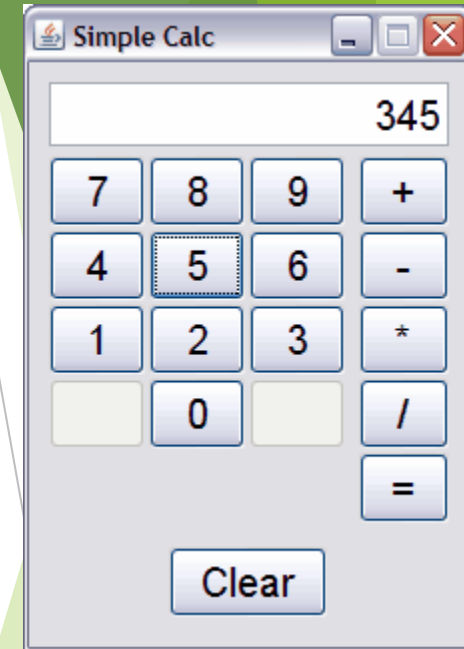
תכונות החשובות

- SPINNER מכיל רשימת הערכים ממנה המשתמש בוחר ערך מסוים.
- SPINNER מאפשר להגיד לכך ערך מוצג, מספר, הנקרא INDEX שבו. משתמשת התוכנה מאחורי הקלעים לאחר בחירת ערך על ידי המשתמש.



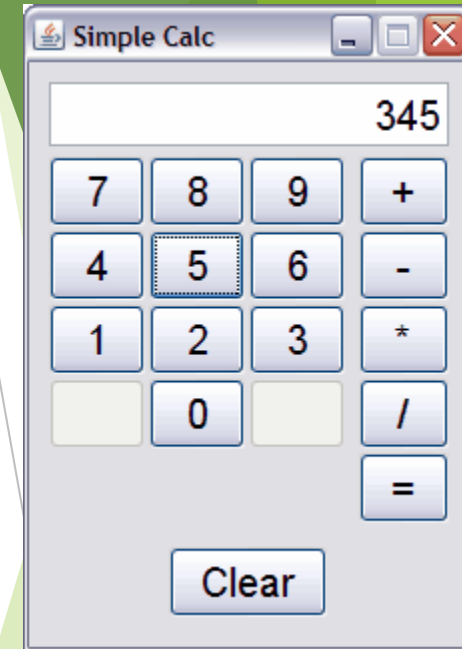
תרגיל מספר 2 - פרויקט מחשבון

- ▶ אנחנו מתחילים לפתח פרויקט הראשון - מחשבון
- ▶ בשלב זה המחשבון צריך להראות ולא לחשב מספר בלבד - המחשבון יכול כפתורים עבור כל ספרה (1,2,3,4,5,6,7,8,9,0)
- ▶ בנוסף יכול המחשבון גם סימנים +, -, *, /
- ▶ וכמובן סימן =
- ▶ יש לסדר את הכפתורים בדומה למחשבון (כדאי להשתמש ב-LAYOUT)
- ▶ מעל הכפתורים יש להציג תיבת טקסט שתציג את התוצאה



תרגיל מספר 2 - פרויקט מחשבון

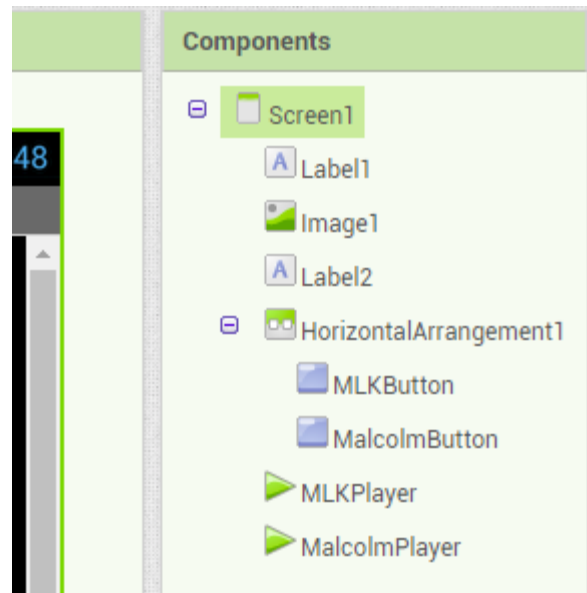
- ▶ בכל לחיצה על הכפתור ספירה, תציג ה TEXTBOX את תו הנלחץ מבלי למחוק את התווים הקודמים
- ▶ בכל לחיצה על הכפתור הסימן, התווים שכבר נכתבו ימחקו
- ▶ חשוב מאוד!!! יש לשמור את האפליקציה (על ידי לחיצה SAVE)
- ▶ חשוב מאוד!! יש לפתח מחשבון ממש ממש פשוט



תרגיל חזרה קצר - פיתוח אפליקציה קיימת

- ▶ פיתוח אפליקציה - יש לי חלום I HAVE A DREAM
- ▶ מכירים את MARTIN LUTHER KING ? למנהיג אמריקאי עבור הזכויות האדם - היה משפט ידוע "יש לי חלום"
- ▶ יש לפתח אפליקציה שמנגנת את המילים הללו
- ▶ יש להשתמש בSTARTER המצורף לקורס 6 lesson - IHaveADreamStarter

תרגיל חזרה קצר - הפתרון



פיתוח אפליקציות

שיעור מספר 4

שיעור 4 - משתנים ולוגיקה

1. הרחבה בנושא BLOCK

2. משתנה

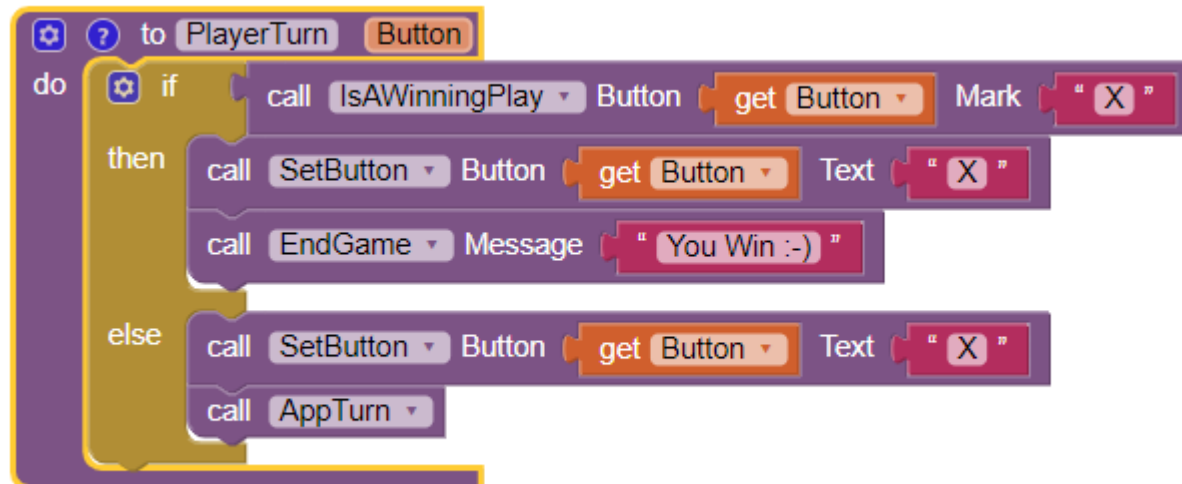
3. תנאים

תרגיל חזרה קצר

- ▶ צור אפליקציה להזנת קוד סודי בכספת
- ▶ הלקוח צריך להציג קוד ובסיום #
- ▶ האפליקציה בכל לחיצה לא מציגה ללקוח את מספר שהוא הזין אלא מציגה כוכביות במקום המספרים
- ▶ לאחר הזנת סולמית ב# הקוד נעלם ומופיע הודעה - "הכספת נעולה"

קצת תזכורת והרחבה לגבי הבלוק BLOCK

- ▶ הלוגיקה כידוע לכם ב APP INVERTOR מתנהלת בתוך חלק BLOCKS בלוקים (BLOCKS)
- ▶ מדוע בעצם BLOCK נקרא כך?
- ▶ כי האפליקציה מחולקת לפקודות והפקודות מתבצעות לאחר תחילת אירוע מסוים..
- ▶ החלק שבו מתבצעת הפקודה מאז שהיא התחילה (מאירוע מסוים) ועד הסיום נקרא BLOCK.
- ▶ בלוק יכול להתחיל לא רק אחרי אירוע מסוים (כמו לחיצה על הכפתור) אלא גם בתחילת התוכנית או כאשר SCREEN - המסך שלנו (נדבר עליו בהמשך) מאותחל (ז"א נטען - נדבר גם על זה בהמשך...)



קצת תזכורת והרחבה לגבי הבלוק BLOCK

initialize global `imageFiles` to `make a list` `"coitTower.jpg"`
`"ggBridge.jpg"`
`"alcatraz.JPEG"`

בלוק שמתחיל
ממש בתחילת
אפליקציה

Show the first picture when app begins

? when `Screen1` .Initialize
do

בלוק שמתחיל כאשר מסך
SCREEN1 אותחל (בשלב זה
יש לנו רק מסך אחד אבל
בהמשך יהיו יותר)

Show the next picture when button clicked

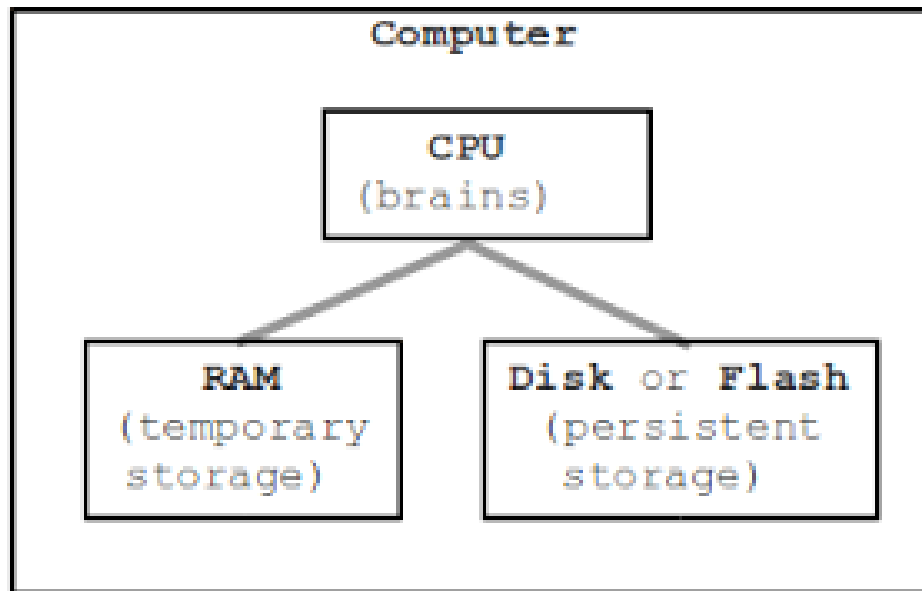
? when `NextButton` .Click
do `set Image1 . Picture to`

כאשר נלחץ כפתור
כלשהו

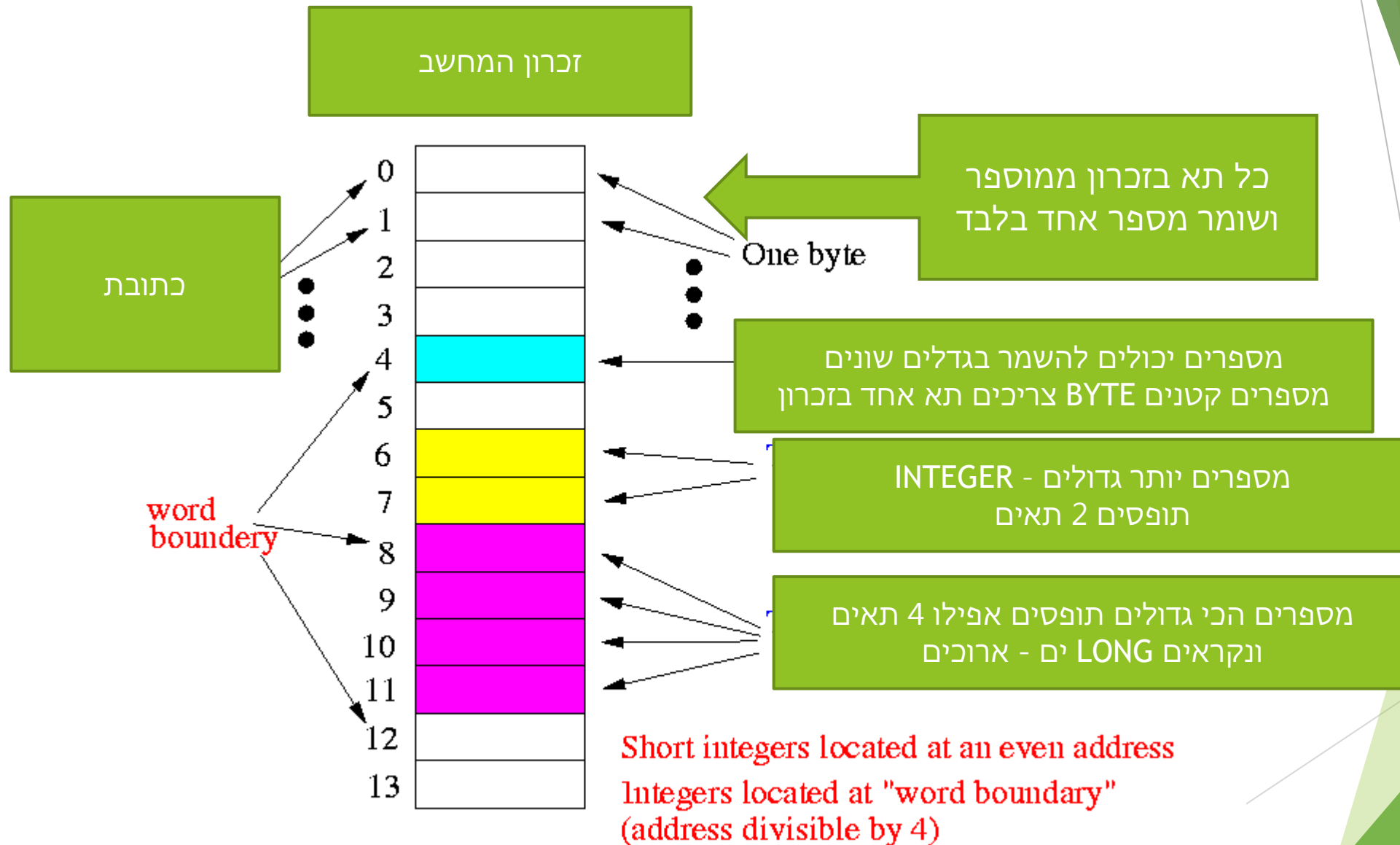
הגיע זמן לחשוב 😊
נדבר על לוגיקה

משתנים

- ▶ מחשב - תפקידו לשמור מידע
- ▶ לצורך שמירת מידע, משתמש המחשב בדיסק ובזכרון
- ▶ בשלב הראשון, כאשר לומדים לתכנת, מדברים על שמירת מידע בזכרון RAM
- ▶ בהמשך נדבר גם על שמירת מידע בדיסק (בסיס נתונים)

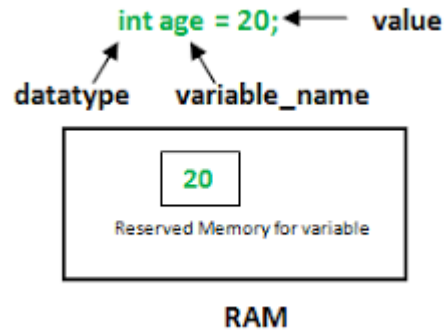


משתנים



משתנים

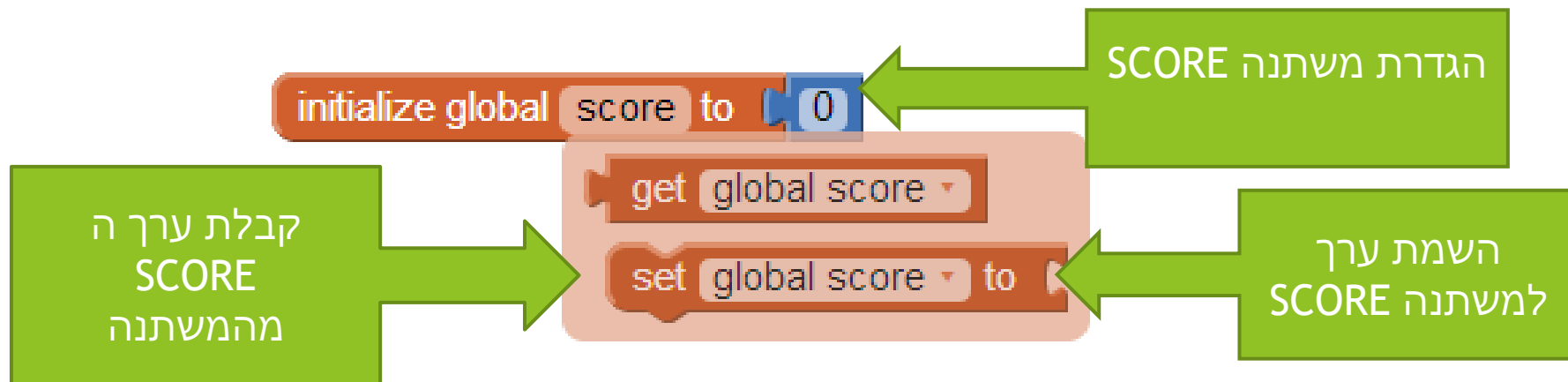
- ▶ משתנה זה בעצם יכול להיות אות או ערך שיכול לקבל ערכים שונים
- ▶ משתנה יכול לקבל ערך מספרי למשל $X=5$
- ▶ משתנה יכול לקבל ערכים טקסטואליים למשל $x="ABCDEF"$
- ▶ בשני המקרים התוכן של המשתמש יוחזק בזכרון
- ▶ ולכן לאחר השימוש במשתנה, ערך שלו ישמר כל עוד האפליקציה עובדת
- ▶ ברגע שהאפליקציה מסתיימת, כל הערכים שלא שמורים בדיסק נמחקים.



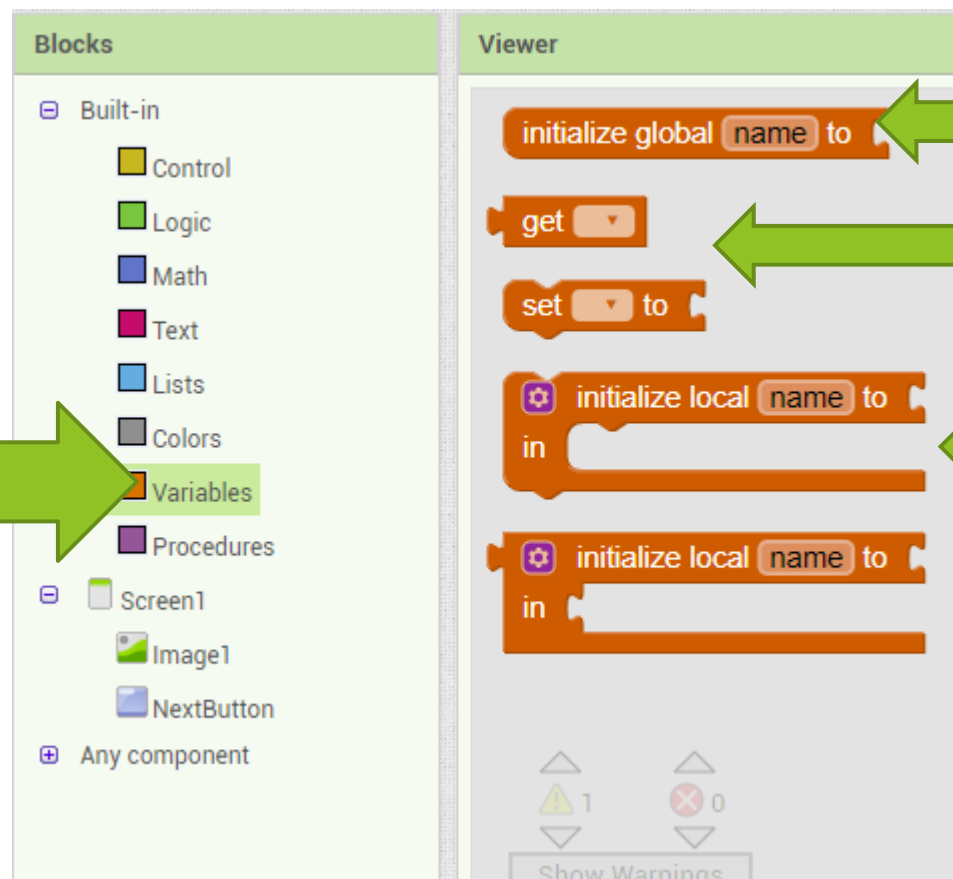
- ▶ כאשר אנחנו מדברים על המשתנים ב APP INVERTOR , אנחנו מתכוונים לכל הגורמים היודעים לנהל ערכים
- ▶ בפקדים ורכיבים - רכיבי TEXT
- ▶ ניתן גם להגדיר משתניים גלובליים ומקומיים
- ▶ כל עבודה עם המשתנים ב APP INVERTOR מתבצעת בחלק BLOCK

משתנים

- ▶ ב-APP INVERTOR המשתנים יכולים להישמר ברמה גלובאלית
- ▶ ז"א הם נשמרים בבלוק הגלובאלי - < שנקרא GLOBAL
- ▶ ברגע שמשתנה נשמר ב-GLOBAL כל הכפתורים והרכיבים יכולים להשתמש בו - ניתן להשתמש ולשנות אותו בכל ה-BLOCKים
- ▶ וגם ניתן לגשת למשתנה בכל BLOCK
- ▶ ניתן לעדכן ערך המשתנה על ידי שימוש ב-SET
- ▶ ניתן למשוך ערך המשתנה על ידי שימוש ב-GET
- ▶ ניתן לאתחל בצורה התחלתית ערך המשתנה על ידי השמה ראשוניים - score=0



בחירת VARIABLES



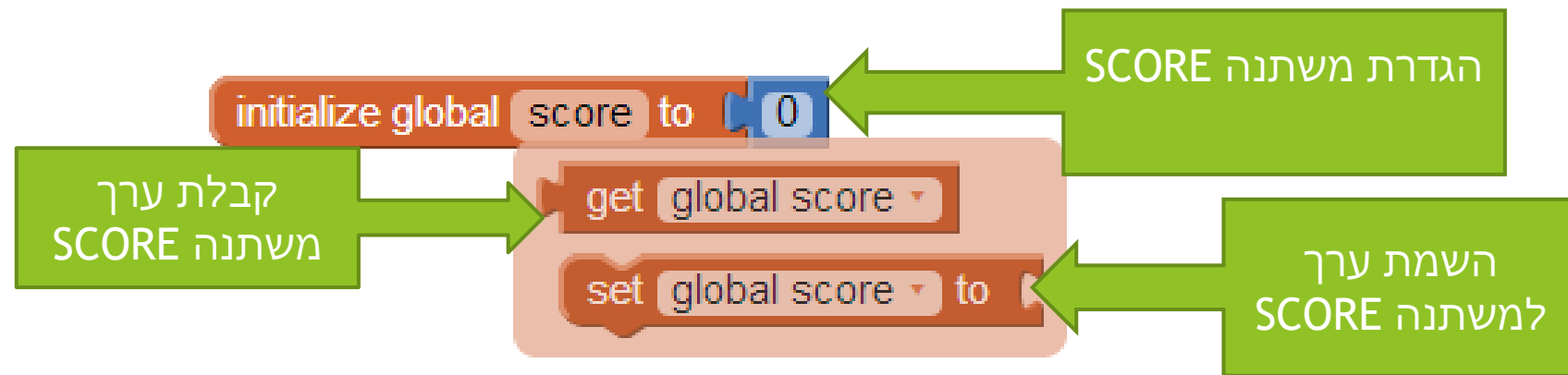
הגדרת משתנה

קבלת ערך מהמשתנה

BLOCK שבו מוגדר
משתנה NAME
בלוק מופעל בלולאה

הגדרת משתנה

משתנים



- ▶ ב APP IVERTOR אינכם יכולים לבחור סוג משתנה אז כיצד המחשב יודע מה סוג המשתנה ??
- ▶ הוא יודע זאת לפי הערך שהעברתם לתוכו
- ▶ למשל אם כתבת `i=4` ... אז המחשב יודע שמעכשיו משתנה בשם `i` יכיל ערך מספרי
- ▶ ואם למשל כתבתם `i="I am a boss"` אז המחשב מבין מייד שמדובר במחרוזת.
- ▶ ואם למשל כתבתם `i=true` אז מעתה `i` יוכל להכיל רק 2 אופציות `true false`
- ▶ חשוב - ברגע שהשמתם ערך לתוך המשתנה, הוא מאז ומעולם יקבע בסוג שקבעתם אותו... למשל אם קבעתם `i=4`, לא ניתן יותר לכתוב `i="ddd"`. כי מבחינת המחשב `i` הוא מספרי

תרגיל מספר 2 - משתנים

- ▶ הגדר משתנה STRING מסוג TEXT s="yes this is my name"
- ▶ הגדר TEXTBOX
- ▶ עדכן את TEXTBOX בערך של STRING
- ▶ הגדר משתנה X
- ▶ עדכן את X במספר 6

משתנים

- ▶ ניתן לשים במשתנה לא רק ערך קבוע, אלה גם משתנה אחר למשל אם i ו j הם מספריים
- ▶ למשל אם קבענו ש $i=1$ אז ניתן לכתוב $j=i$ וגם j יהיה שווה ל 1
- ▶ אם קבענו למשל ש $i=8$ ניתן לכתוב גם $i=i+1$ ואז ערך של i יהיה 9
- ▶ כמובן, בדיוק כפי שכתבנו קודם, לא ניתן לשים משתנה מספרי למשתנה טכסטואלי ולהיפך

משתנים

- ▶ על מנת לשים משתנה מספרי ב APP INVERTOR נדרש למשתנה ב MATH
- ▶ MATH עובד אף ורק עם המספרים
- ▶ תפקידו של MATH לא רק לשים ערכים מספריים אלה גם לבצע פעולות חשבוניות שדיברנו עליהם בשקף הקודם
- ▶ ולא רק פעולות פשוטות, אלה גם מורכבות ואפילו מורכבות מאוד

משתנים

The image displays the Scratch interface with a focus on the 'Variables' category in the 'Blocks' panel. The 'Viewer' panel shows a sequence of variable-related blocks: '0', '=', '+', '-', 'x', '/', '^', and 'bitwise and'. Green arrows point from Hebrew labels to these blocks. To the right, three code snippets illustrate the use of a local variable 'x1'.

Blocks Panel:

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
- Image1
- NextButton
- Any component

Viewer Panel:

- 0
- =
- +
-
- x
- /
- ^
- bitwise and

Labels (from top to bottom):

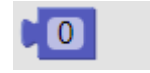
- השמת מספר כלשהו למשתנה
- חיבור בין 2 משתנים
- חיסור בין שני המשתנים
- כפל בין שני המשתנים
- חילוק בין 2 המשתנים
- חזקה

Code Snippets:

- Snippet 1: initialize local x1 to 222
- Snippet 2: initialize local x1 to 222, in set x1 to 333 + 222
- Snippet 3: initialize local x1 to 222, in set x1 to 333 - 111

משתנים

דגשים לעבודה עם MATH



בכל שימוש במספר יש להשתמש ב

בכל פעולת השמה נדרש להשתמש בSET

ז"א אם רוצים לכתוב למשל $x=10+20$



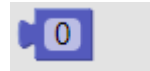
נדרש לגרור תחילה את

לאחר מכן נדרש לבחור בתיבת בחירה את X

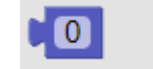


לאחר מכן נדרש לגרור את פעולת החיבור לתוך סוף הבלוק

לאחר מכן יש למקם את 10 הראשון תוך שימוש ב



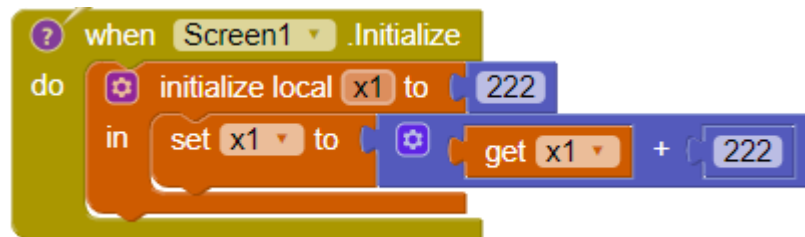
ולאחר מכם את 20 תוך שימוש



ב

אם אנחנו רוצים במקום 10 או במקום 20 לשים ערך של משתנה יש לגרור את GET של

מהמשתנה למשל אם רוצים לכתוב $X=X+222$

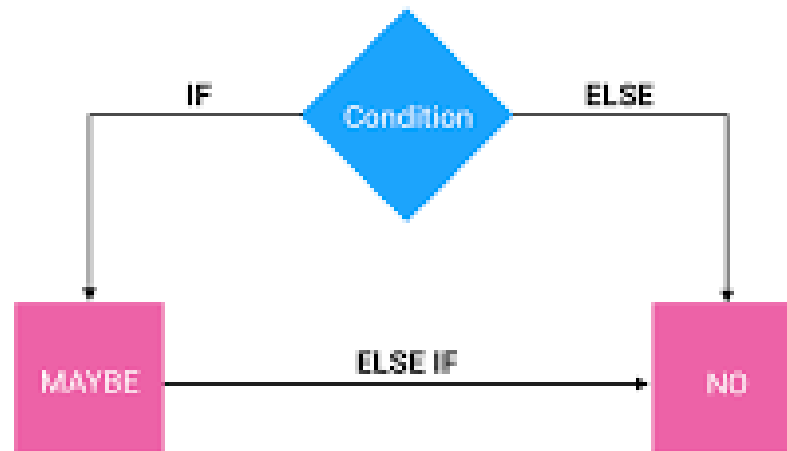


תרגיל מספר 3 - משתנים

- ▶ כתוב אפליקציה לצורך ספירת כניסות למחיאון. השומר אשר מציגה תחילה את הכפתור ומספר 0 ובלחיצה על הכפתור, מגדילה את מספר ב1. כפתור נוסף מאפשר את החשבון חזרה 0
- ▶ כתוב אפליקציה אשר מקבלת 2 מספרים. האפליקציה מדפיסה ב4 תיבות טקסט את הפעולות כפל, חיבור, חילוק וחסור של 2 המספרים הללו
- ▶ כתוב אפליקציה אשר קולטת מספר ובכל לחיצה מדפיסה את החזקה שלו פי 2

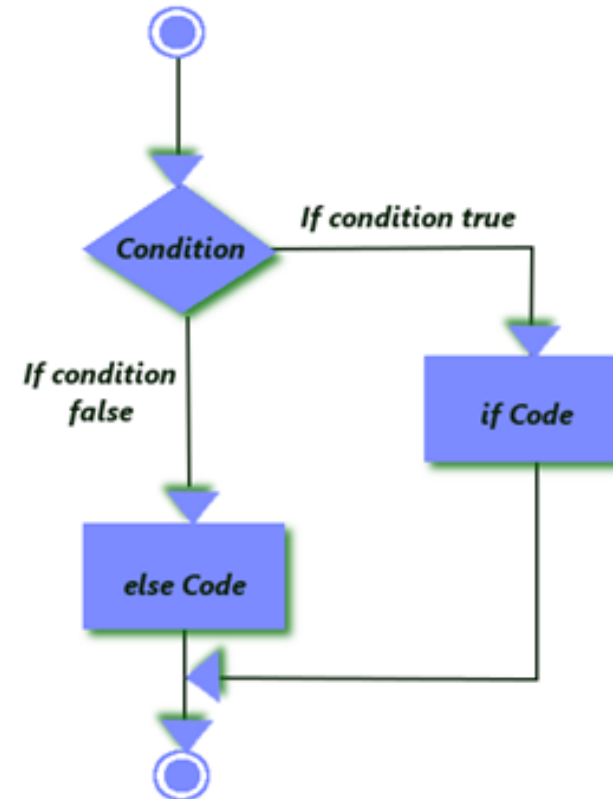
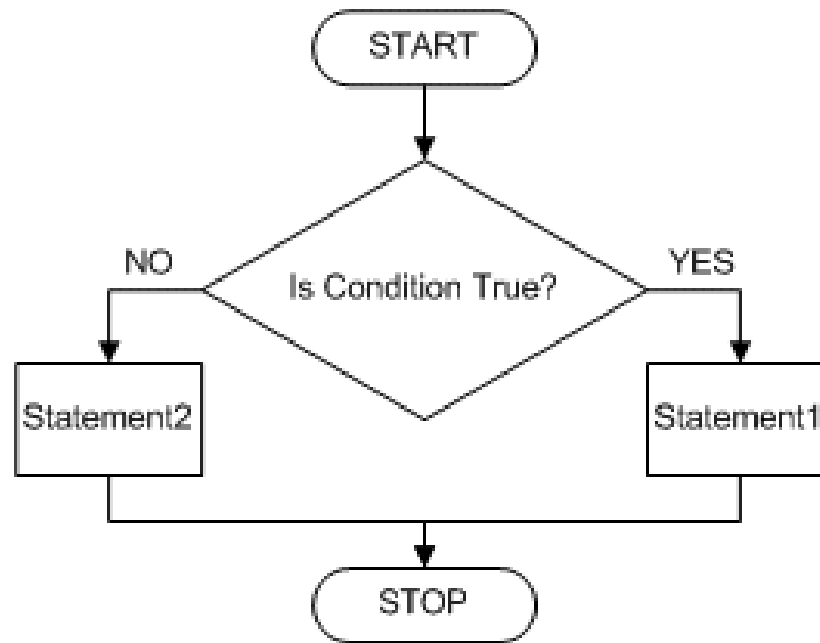
תנאים

- ▶ דבר נוסף שחשוב מאוד להבנת הלוגיקה של מחשב הוא **תנאי**.
- ▶ תנאי זאת ההחלטה שיש לאפשר למחשב מה לבצע, בהתאם לערך המשתנה שלו.
- ▶ תנאי מאפשר לאפליקציה להיות דינמית ולהתאים את עצמה למשתמש
- ▶ בשפות תכנות, שיתכן חלקכם מכירים, תנאי מכונה וידוע תחת השם IF.



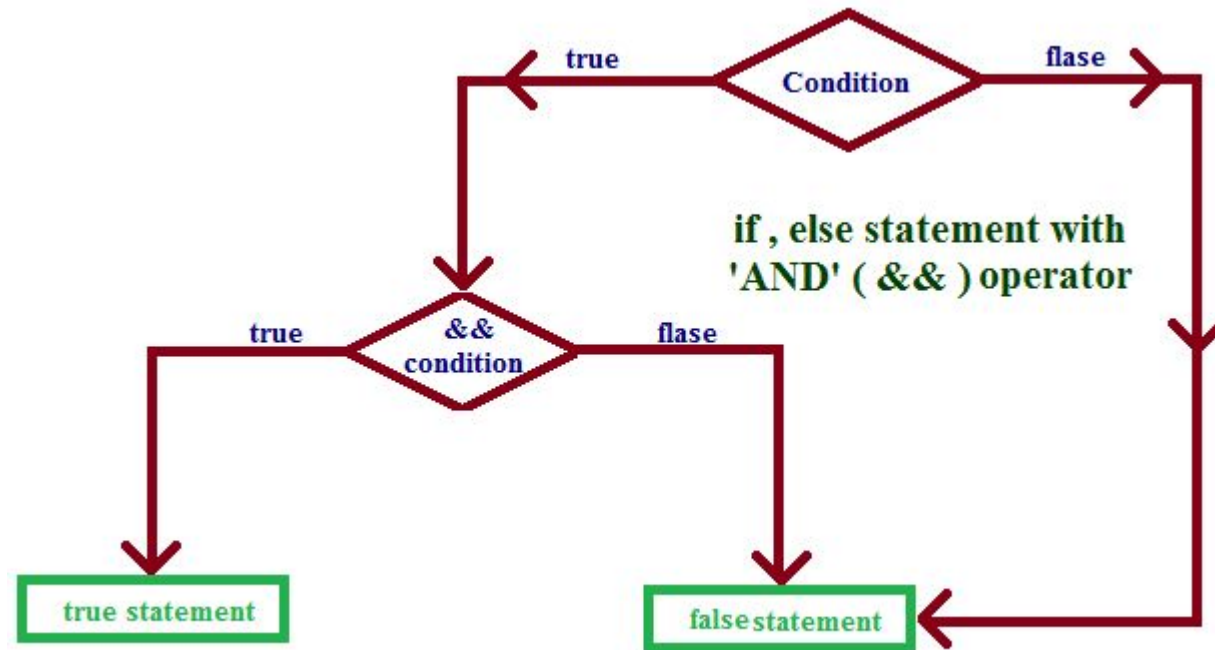
תנאים

- ▶ לדוגמא בתרשים שמופיע למטה, מחשב בוחר תנאי.
- ▶ במידה והתנאי הוא אמת מתבצע חלק מהתוכנה תחת YES ובמידה שקר - מתבצע תחת NO



תנאים

- ▶ ניתן לחבר את התנאים
- ▶ ניתן ליצור תנאי אחרי תנאי וכך הלאה , ניתן ליצור תנאי מורכב
- ▶ ניתן גם לשאול ומה אם לא - if - else if



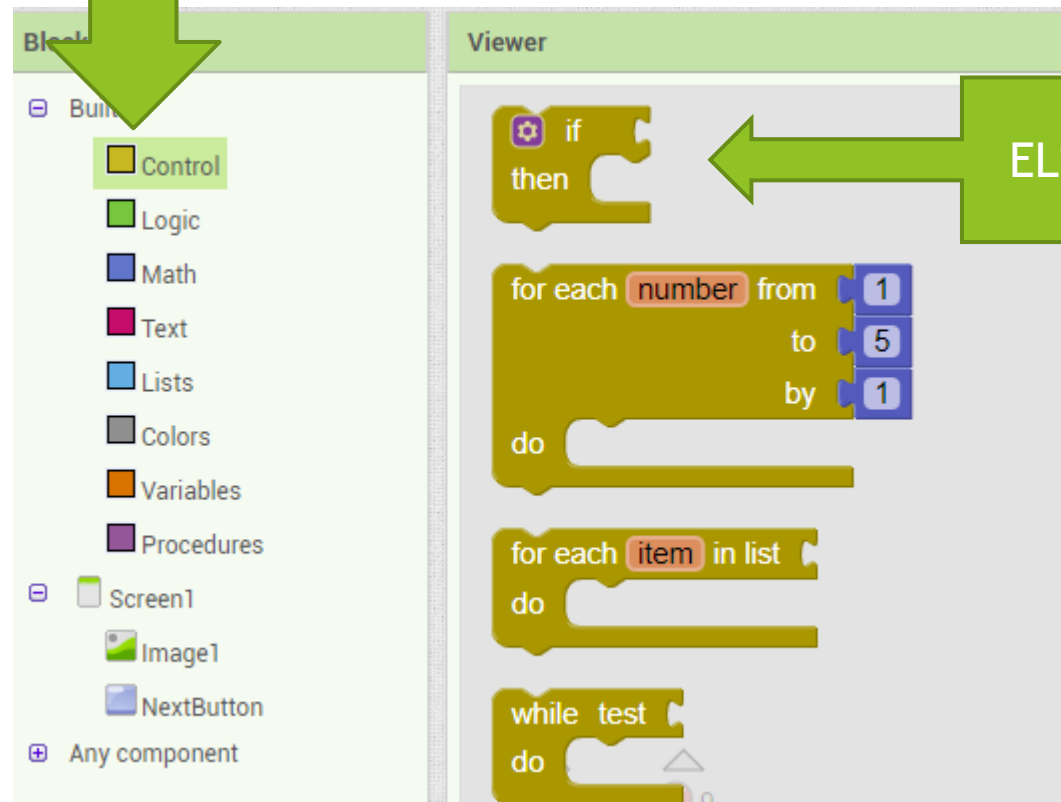
תנאים

ניתן לחבר את התנאים

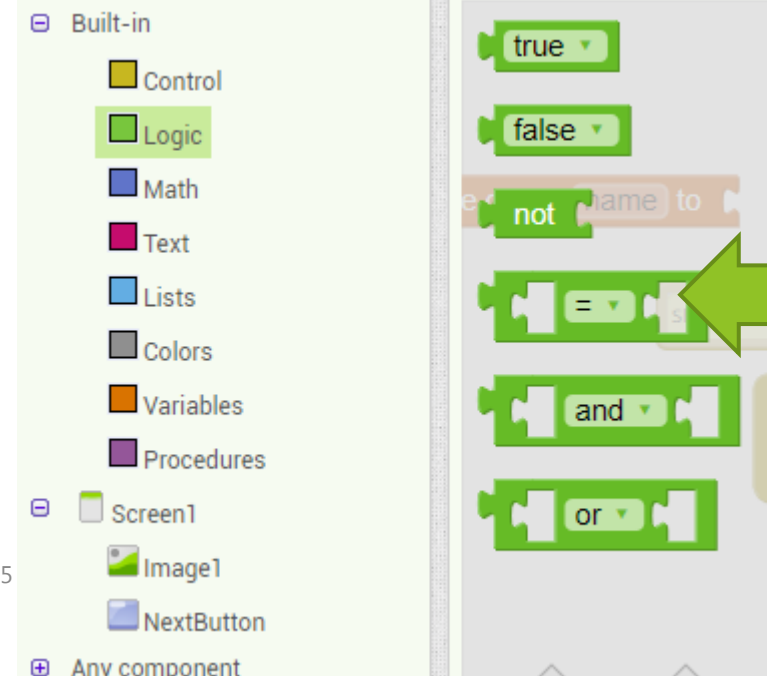
ניתן ליצור תנאי אחרי תנאי וכך הלאה , ניתן ליצור תנאי מורכב

ניתן גם לשאול ומה אם לא - if - else if

בוחרים
באופציה בקרה
CONTROL -



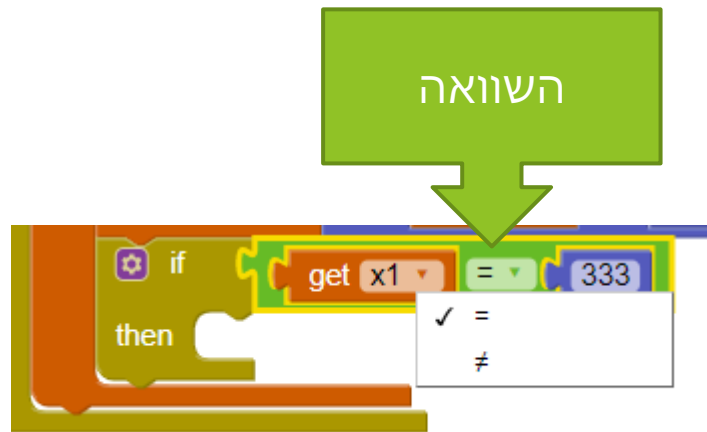
גוררים את IF וELSE



לאחר מכן
גוררים את
התנאי

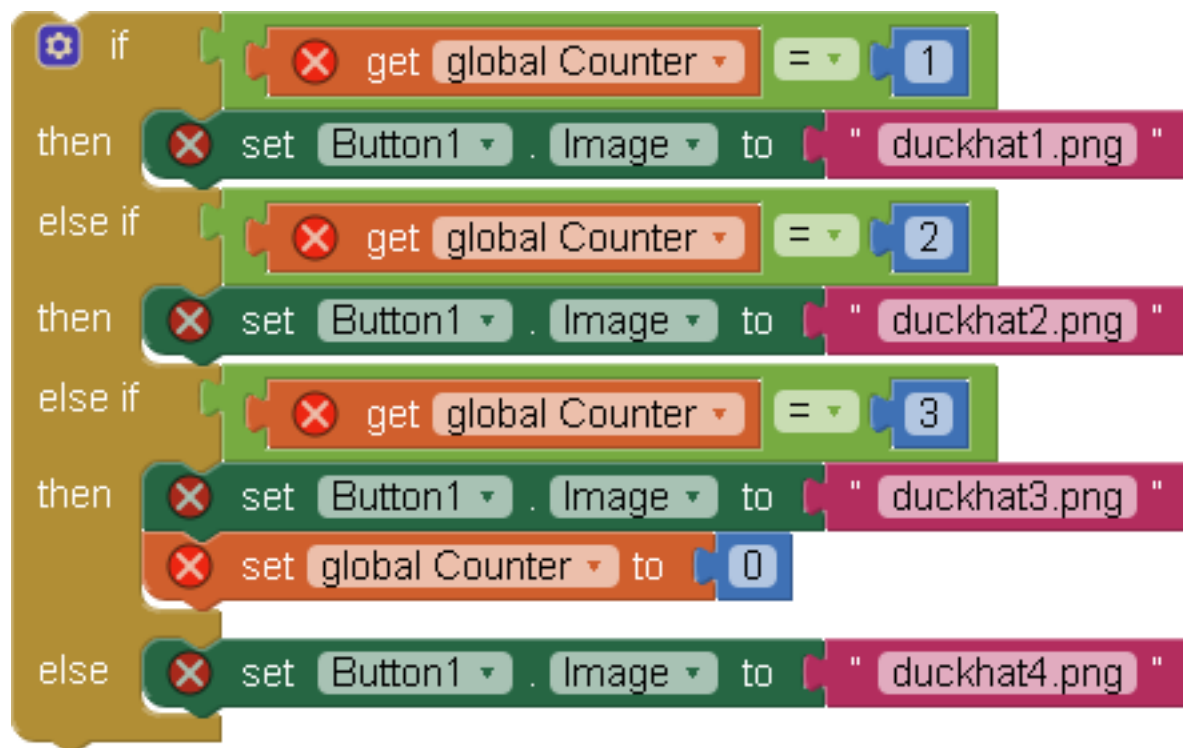
תנאים

- ▶ כאשר גוררים תנאי אפשר ללחוץ על סימן ולבחור בסימן לוגי רצוי
- ▶ אנחנו יכולים לא רק להשוואת בין מספר למספר , אלה לבדוק האם מספר אחד גדול מהשני, קטן מהשני ועוד

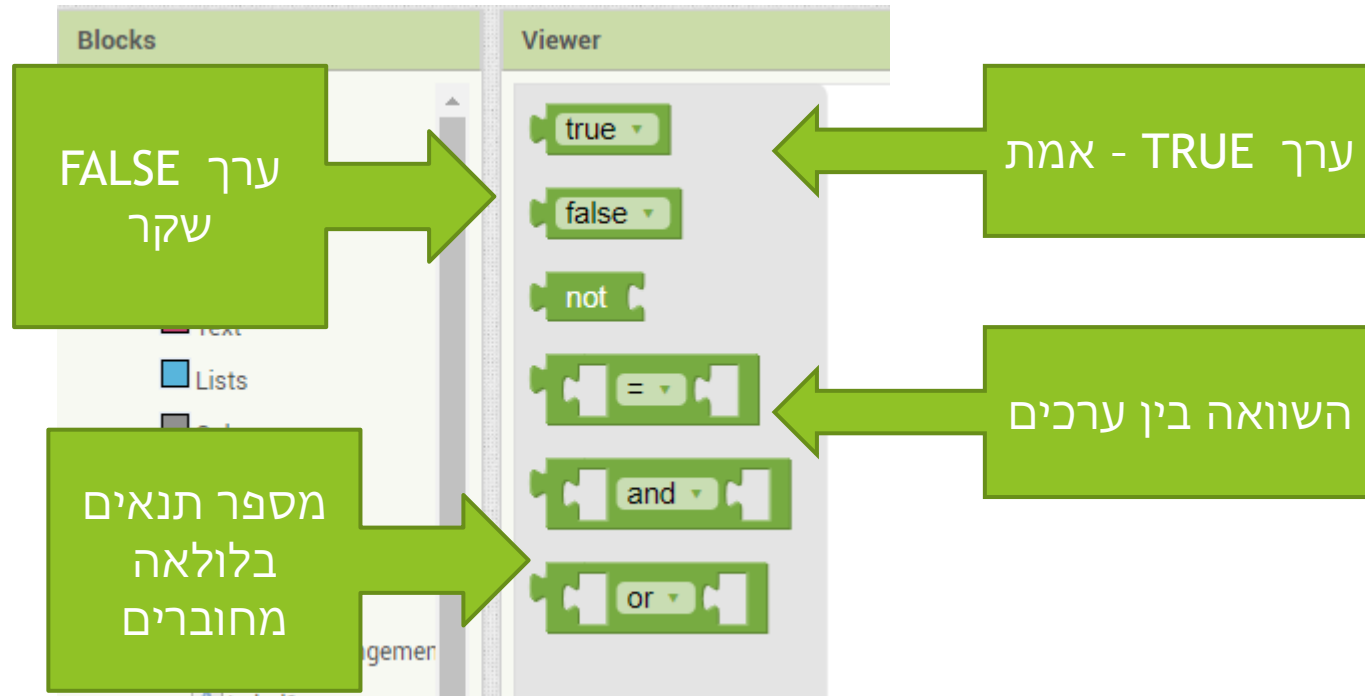


תנאים

- ▶ בתנאי הבא אנחנו מודדים ערך משתנה
- ▶ אם ערך שווה ל1 אז מעדכנים את התמונה לשם א
- ▶ אם ערך שווה ל2 אז מעדכנים את התמונה לשם ב וכך הלאה
- ▶ נדבר על התמונות במצגות הבאות



תנאים

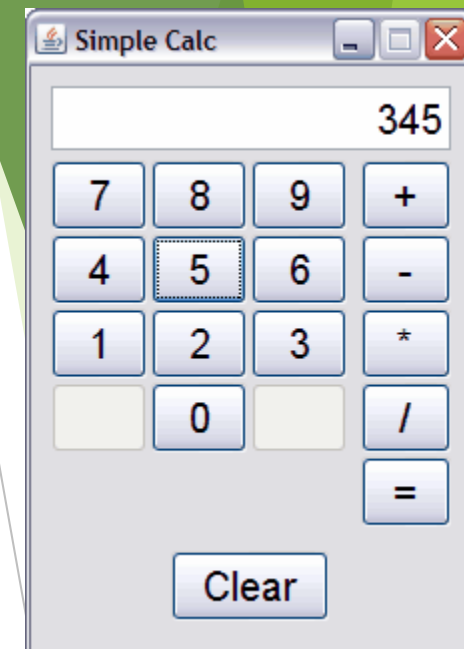


תרגיל מספר 4 - תנאים

- ▶ כתוב אפליקציה שמקבלת 6 מספרים ומדפיסה את הסכום שלהם והאם הסכום גדול מ1000
- ▶ כתוב אפליקציה אשר קולטת 2 מספרים ומדפיסה איזה מהמספרים הכי גדול
- ▶ כתוב אפליקציה שבודקת האם מספר מתחלק ב2. (איך עושים זאת??)

תרגיל מספר 4 - המשך מחשבון

- יש לפתוח את אפליקציה המחשבון שפיתחתם בשיעור הקודם.
- עכשיו יש לממש את הפעולת חיבור על ידי לוגיקה הבאה:
- תניחו שהמשתמש לוחץ על הכפתורים בסדר הבא
- מספר ראשון
- סימן
- מספר שני
- שווה
- כאשר המשתמש לוחץ על מספר הראשון - יש לשמור אותו במשתנה גלובלי FIRST
- כאשר המשתמש לוחץ על סימן יש לשמור אות במשתנה גלובלי SIGN
- כאשר המשתמש לוחץ על מספר השני - יש לשמור אותו במשתנה גלובלי SECOND
- כאשר המשתמש לוחץ על שווה - יש לבצע חישוב יש לעשות IF ולבדוק למה שווה SIGN ובהתאם לבצע חישוב ולהציג את התוצאה



פיתוח אפליקציות

שיעור מספר 5

שיעור 5 - מחרוזות ותמונות

1. המשך המשתנים - מחרוזות בBLOCKS
2. פעולת JOIN
3. רשימה LISTS
4. תמונה
5. טעינת וניהול קבצים
6. פיתוח מצגת ראשונה
7. מעבר על אפליקציה קיימת

המשך פרויקט ראשון - המשך המחשבון

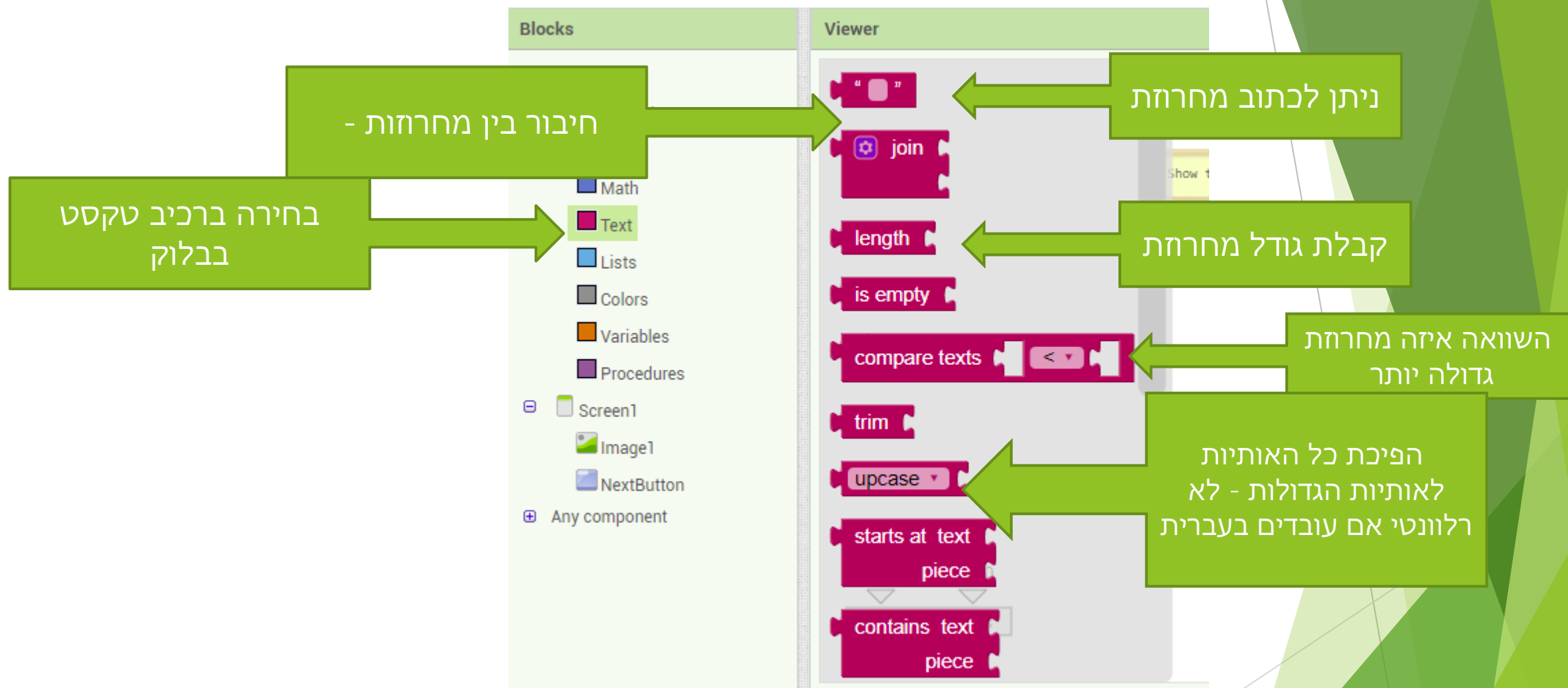
- ▶ המשך לפרויקט המחשבון מהשיעור הקודם.
- ▶ מי שלא סיים - להמשיך את הפרויקט
- ▶ מי שכן סיים - להוסיף כפתור חזקה " $^$ "
- ▶ אם המשתמש לוחץ על מספר
- ▶ אחר כך על חזקה
- ▶ אחר כך על עוד מספר
- ▶ ואחר כך על שווה
- ▶ יש להציג את התוצאה של אותו מספר בחזקה של מספר השני
- ▶ כמובן יש לשמור את האפליקציה

משתנה טכסט שימוש בטקסט בBLOCKS

- ▶ הרבה פעמים, נדרשים אנו להציג ערך טקסטואלי
- ▶ על מנת לעשות זאת אנו יכולים להשתמש בערך TEXT בBLOCKS
- ▶ APP INVERTOR מאפשר לנו לבחור ולבצע מספר פעולות על TEXT
- ▶ תוכניתן (שזה אתם כמובן 😊) יכול לבצע פעולות שונות על טקסט - למשל ניתן לחבר בין 2 מחרוזות

$$AA+BB= AABB$$

שימוש בטקסט בBLOCKS

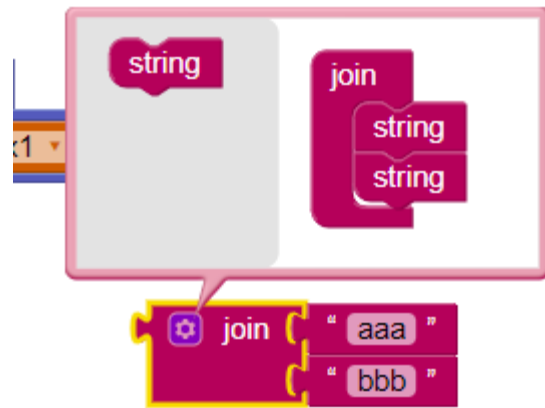


משתנה טכסט - JOIN

▶ פעולת JOIN מחברת בין 2 מחרוזות

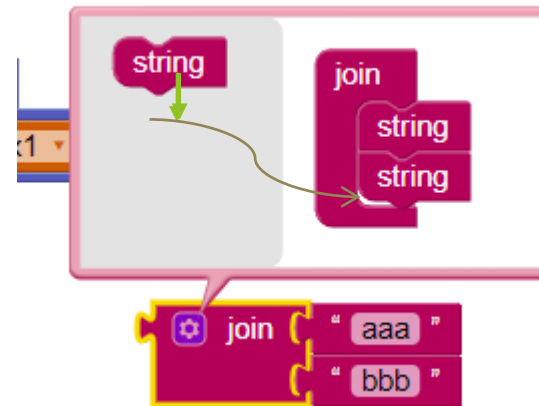
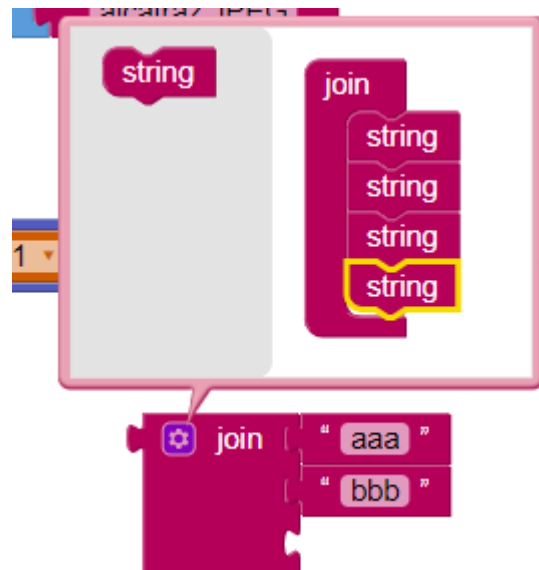
▶ כמובן התוצאה של פעולה הזאת תהיה

משתנה שערכו aaabbb



משתנה טכסט - JOIN - הרחבת פעולה

- ▶ ניתן להרחיב את הפעולה - הרחבת פעולה בAPP INVERTOR מתבצעת על ידי לחיצה על הכפתור הכחול
- ▶ לאחר הלחיצה, בתיבה שנפתחת, תלחצו על הSTRING ותגררו אותו לתוך JOIN העליון... ניתן לגרור כמה STRINGים שרוצים!!!

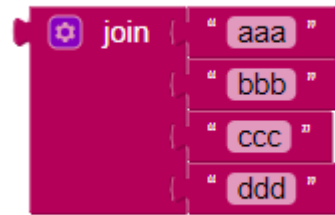


תרגיל מספר 1

- יש לפתח אפליקציה 1 חיבור למשפט
 - "אני" + "תוכניתן" + "אפליקציות" + "תותח"
 - יש להציג את משפט בלחיצה על הכפתור
 - רגע... לא שכחתם "רווחים" 😊
-
- יש לפתח אפליקציה 2 שמציגה 2 תיבות טקסט וכפתור
 - בכל לחיצה המחרוזות שהוקשה בתיבת טקסט ראשונה מתוספת לסוף של מחרוזת טקסט השניה

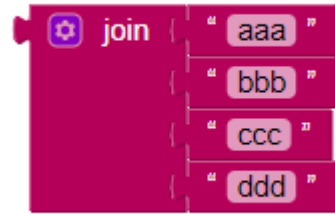
משתנה טכסט - JOIN - הרחבת פעולה

▶ לאחר הגרירה ניתן לבנות ממש מחרוזות ארוכים



משתנה טכסט - JOIN - הרחבת פעולה

▶ לאחר הגרירה ניתן לבנות ממש מחרוזות ארוכות



רשימה - LISTS

- ▶ רשימה זה משתנה שיכול להכיל בו זמנית כמה משתנים
- ▶ נניח למשל שאת רוצים לשמור 100 משתנים ... האם תגדירו 100 משתנים בנפרד?
- ▶ ומה עם 1000 ? 10000 ? 100000 ?
- ▶ בדיוק לצורך זה נוצרה הרשימה LISTS

רשימה - LISTS

- ▶ ניתן ליצור רשימה ריקה
- ▶ ניתן ליצור רשימה שכבר מכילה ערכים
- ▶ ניתן להוסיף ערכים לרשימה קיימת

The image shows the 'Lists' block palette in the Scratch software. On the left, a green box labeled 'בניית רשימה קיימת' (Building existing list) has an arrow pointing to the 'make a list' block. Above the 'create empty list' block, a green box labeled 'יצירת רשימה ריקה' (Creating empty list) has an arrow pointing to it. To the right of the 'add items to list' block, a green box labeled 'הוספת ערכים לרשימה קיימת' (Adding items to existing list) has an arrow pointing to it. The 'Lists' category is selected in the left sidebar, and various list-related blocks are visible in the main palette.

בניית רשימה קיימת

יצירת רשימה ריקה

הוספת ערכים לרשימה קיימת

create empty list

make a list

add items to list list item

is in list? thing list

length of list list

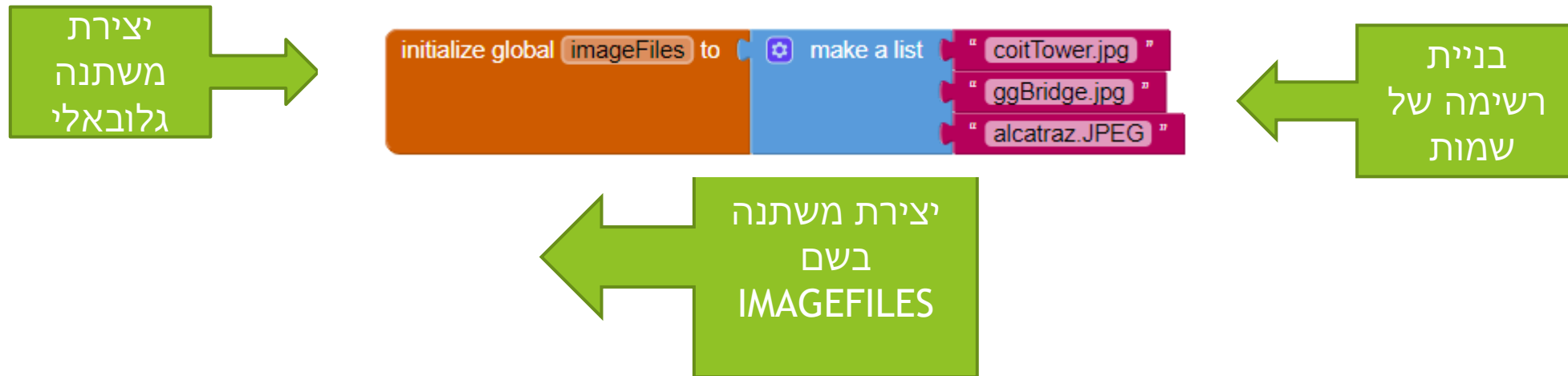
is list empty? list

pick a random item list

index in list thing list

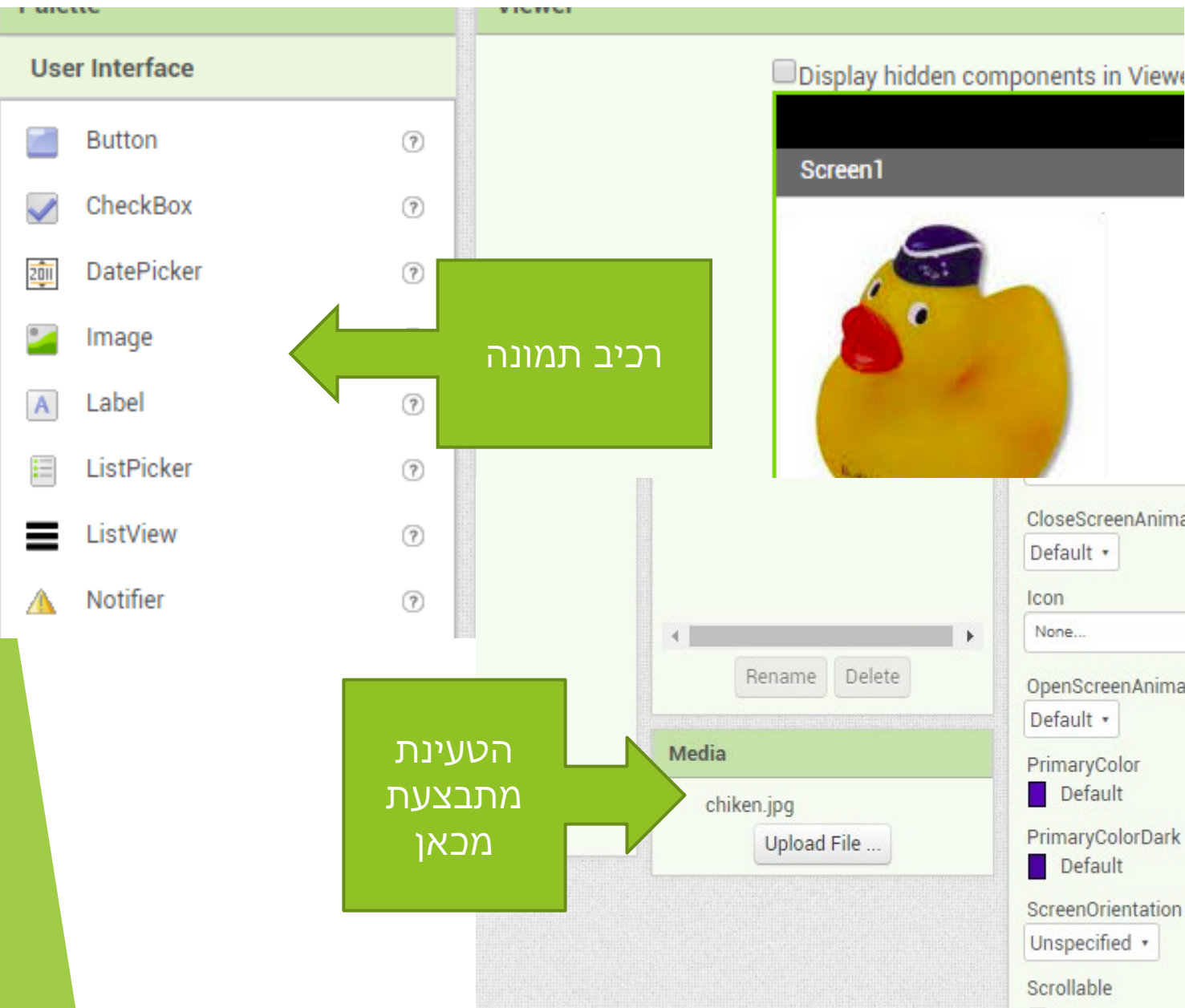
רשימה - LISTS

- ▶ ניתן ליצור משתנה LISTS כרשימה גלובאלית
- ▶ בכך ה LISTS יהיה זמין בכל הבלוקים של תוכנית



- ▶ ניתן להציג כל תמונה
- ▶ חשוב שהתמונה תתאים לגודל החלון באפליקציה
- ▶ ניתן להציג את התמונה ביותר מחלון אחד
- ▶ חשוב להבדיל בין הצגת התמונה לטעינת התמונה באפליקציה
- ▶ ניתן להציג תמונה ממש לא רק על פני רכיב תמונה אלה גם על פני הכפתורים ועוד....

טעינת תמונה



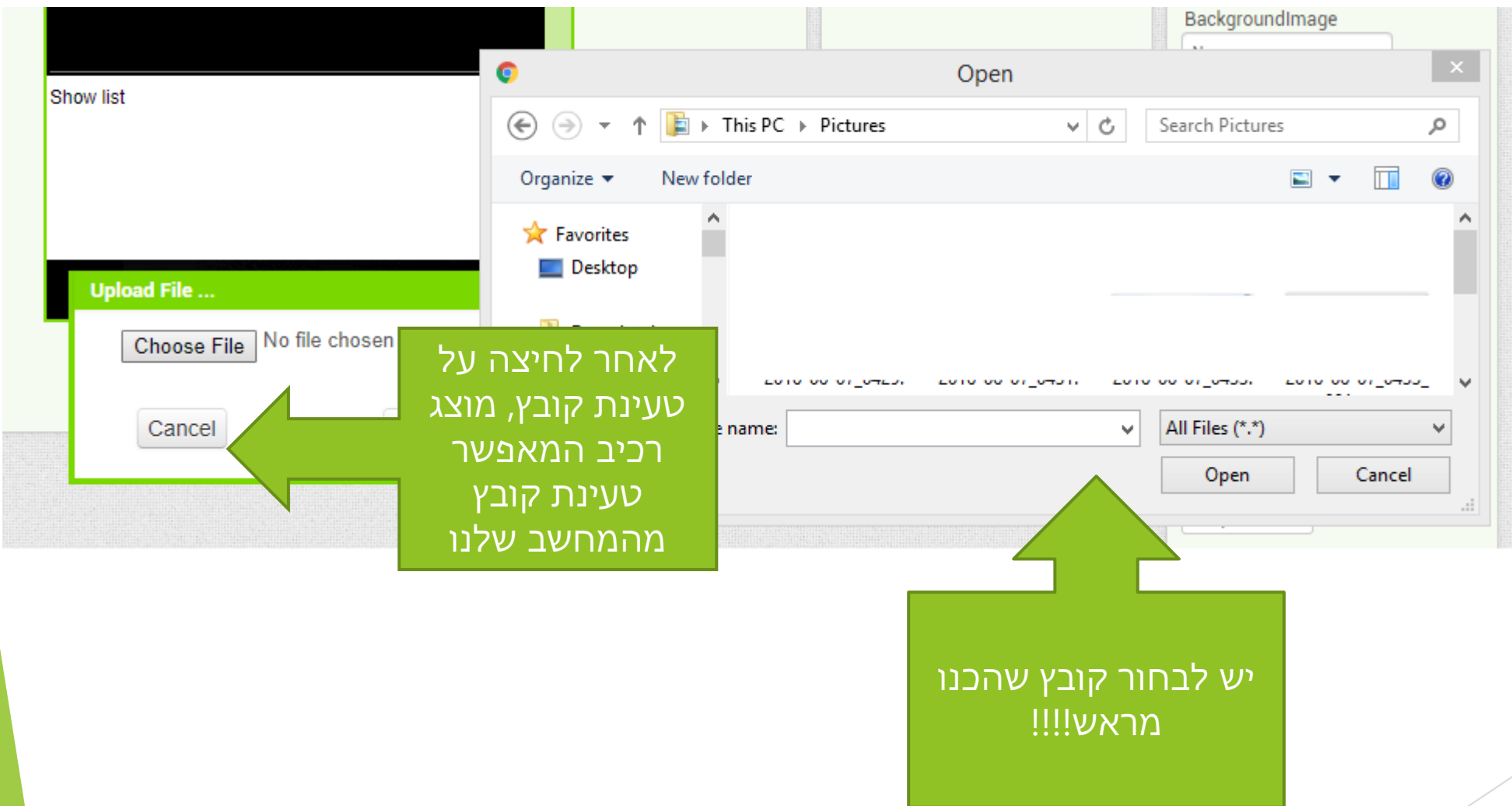
לפני שמציגים את התמונה

יש לטעון אותה לאפליקציה

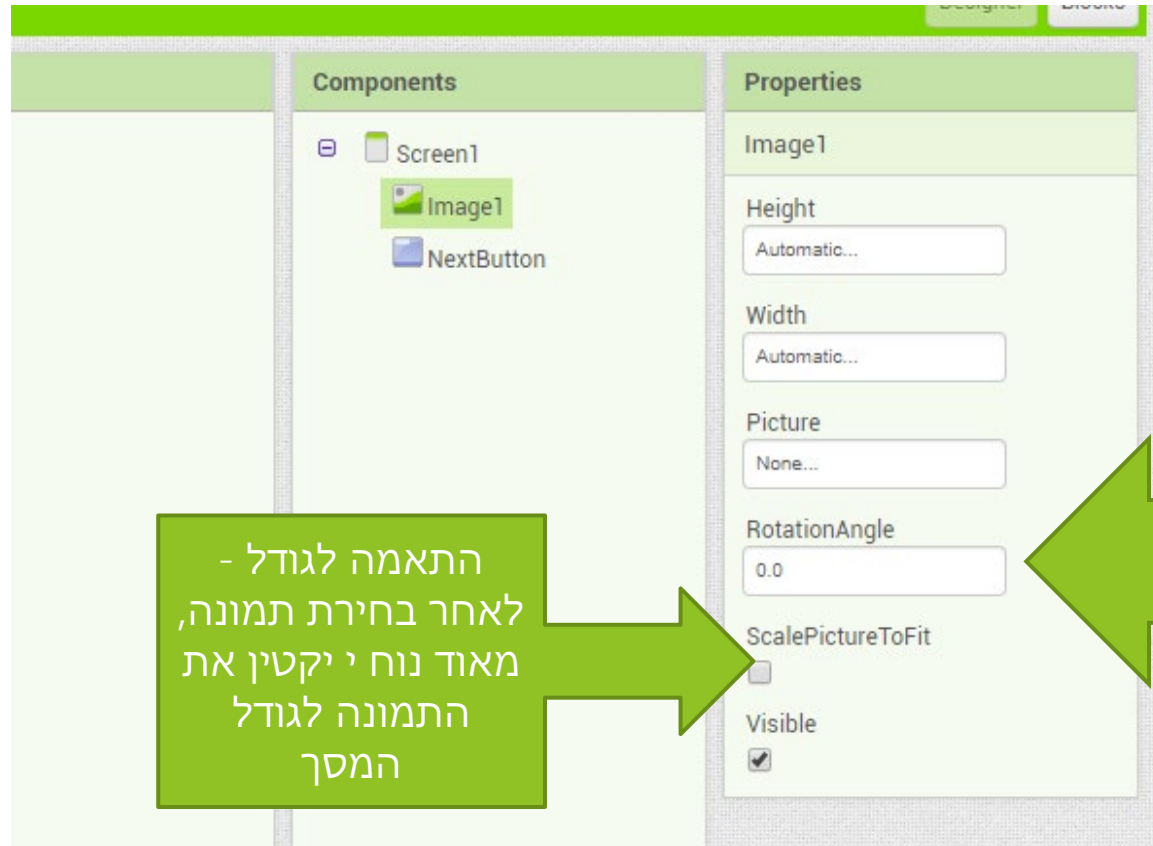
לא חייבים להשתמש בתמונות שטענו

(אבל כמובן לא כדאי להחזיק באפליקציה תמונות מיותרות)

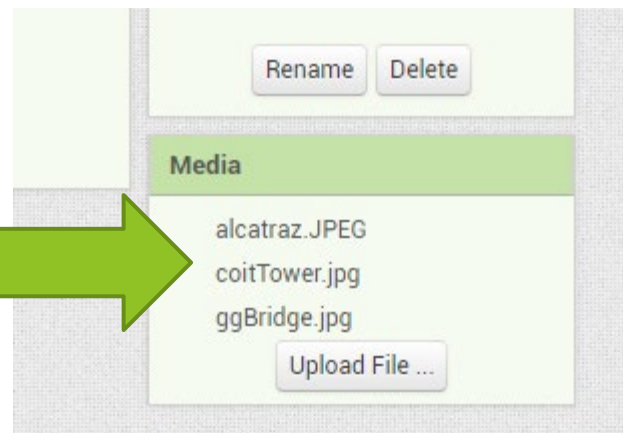
טעינת התמונה



רכיב IMAGE



לרכיב תמונה יש תכונה
PICTURE
כאן ניתן לבחור את שם
התמונה המוצגת
כפי שדיברנו יש לטעון את
התמונות מראש!!



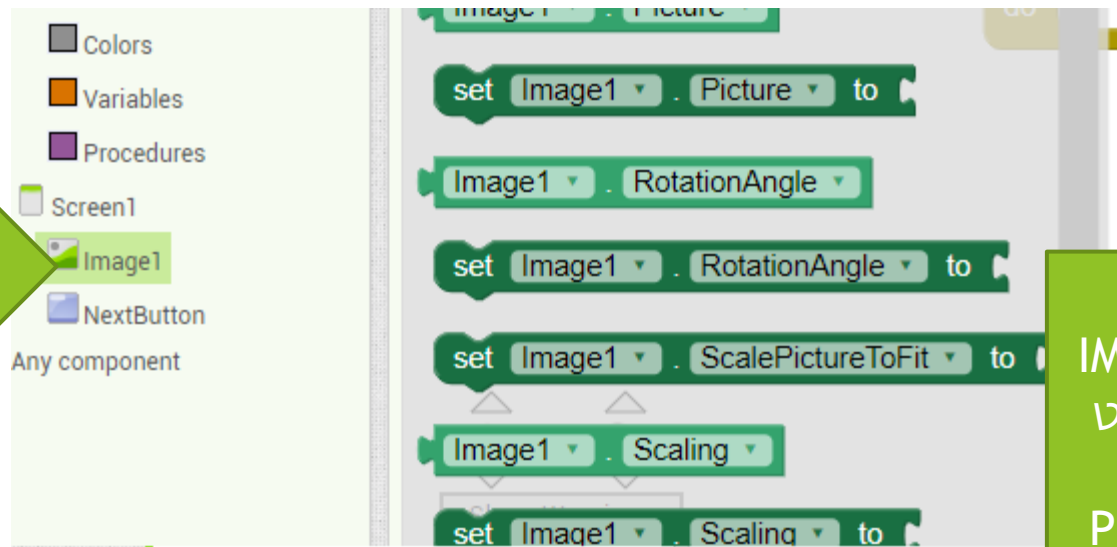
ניתן לבחור רק את
התמונות הללו להיות
מוצגים

שימוש בIMAGE בחלק של BLOCKS

- ▶ לאחר שהכנו מספר תמונות מראש, ניתן לקבוע מתי להציג אותן
- ▶ נניח רצינו להציג את התמונה בלחיצה על הכפתור
- ▶ נעבור לחלק של BLOCK
- ▶ על מנת לעדכן את התמונה, ניתן להשתמש בתיבת טקסט...

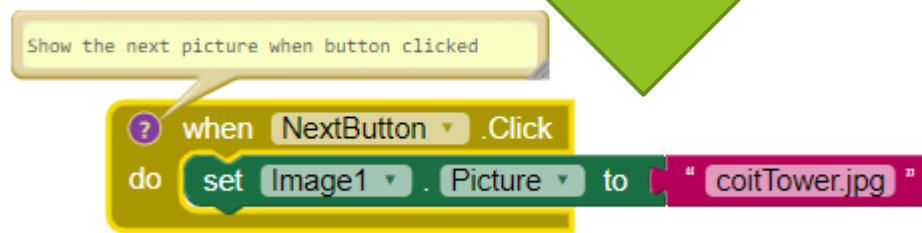
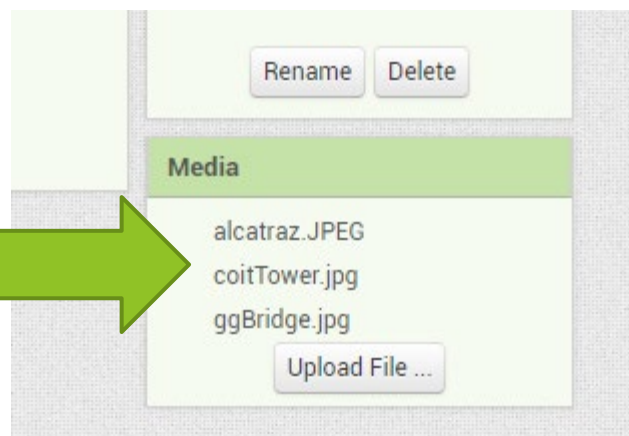
שימוש ועדכון IMAGE בחלק של BLOCKS

ניתן לבצע פעולות
שונות על התמונה



יש לעדכן את ערך שדה IMAGE
על ידי השמה של שדה טקסט
המכיל את שם הקובץ
יש לבצע SET לערך PICTURE
של IMAGE עם שם הקובץ

ניתן לבחור רק את
התמונות הללו
להצגה



עשה ועל תעשה בבחירת קבצים

- ▶ כאמור יש להכין את הקבצים מראש
- ▶ בטעינת יש לקחת בחשבון את גודל הקבצים - כל קובץ גדול מגדיל לכם את גודל האפליקציה - < ולכן גם מגדיל זמן שיקח לאחר מכן למשתמש להוריד אותה למחשב!!
- ▶ יש לקחת בחשבון את גדלי המכשירים שבהם משתמשים המשתמשים
- ▶ לא תמיד אותה תמונה שמתאימה לסמסונג 15A מתאימה לטאבלט....
(אולי אפילו צריך לטעון כמה תמונות מראש)

תרגיל מספר 2 - הצגת תמונות

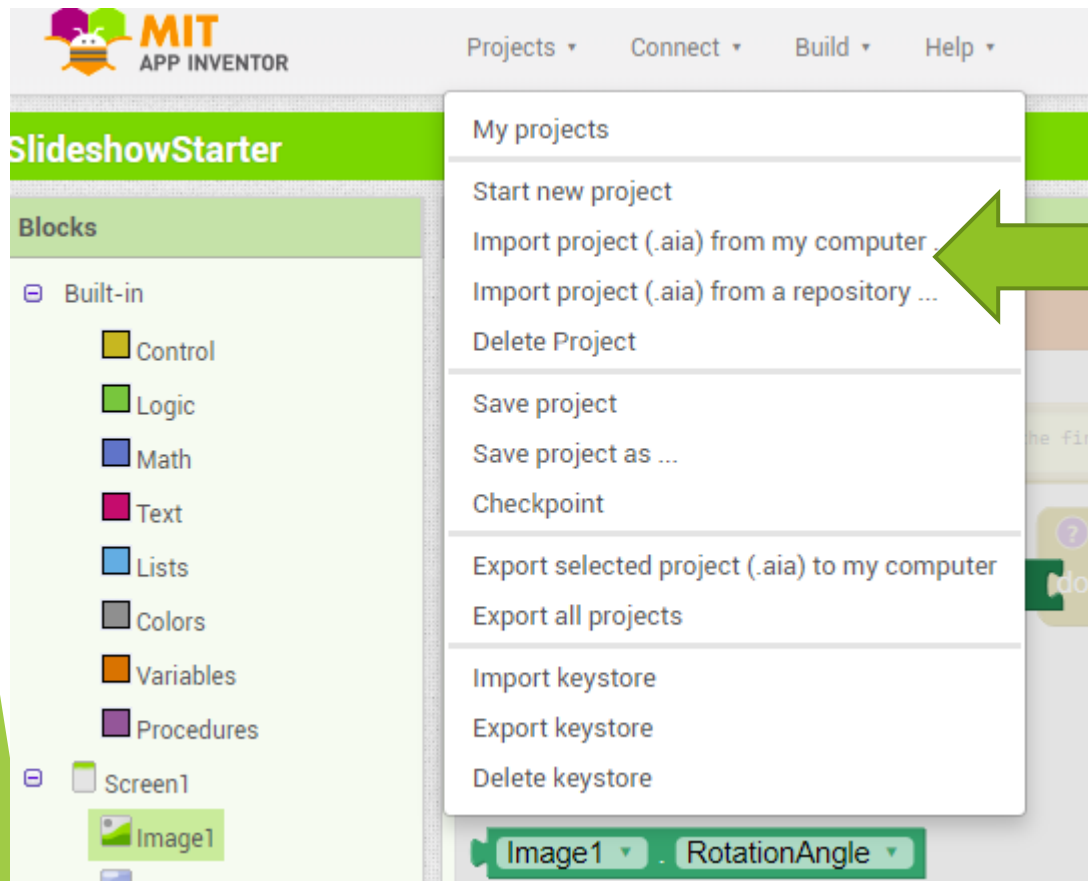
- ▶ יש ליצור אפליקציה המציגה 3 תמונות (אשר הכנתם מראש)
- ▶ יש להכין מסך עם התמונה (כאשר מראש מוצגת תמונה ראשונה מתוך 3 - ניתן להציגה מעל רכיב IMAGE או כפתור)
- ▶ יש להוסיף למסך זה 3 כפתורים
- ▶ בלחיצה על כל כפתור, תתחלף התמונה המוצגת ותוצגנה תמונה הבאה.

טעינת אפליקציה קיימת מהמחשב..

- ▶ לא חייבים תמיד לפתח הכל מ0
- ▶ ההיפך הוא הנכון.. רב התוכניתנים תמיד משתמשים בתוכנות וברכיבים קיימים
- ▶ ניתן להוריד אפליקציות קיימות ולהשתמש בהם ולהרחיב אותן

טעינת אפליקציה קיימת מהמחשב..

- ▶ על מנת לטעון לAPP INVERTOR אפליקציה קיימת, יש להכין את הקובץ במחשב (להוריד אותו)
- ▶ למשל הכנו לכם קובץ של אפליקציה מסוימת...



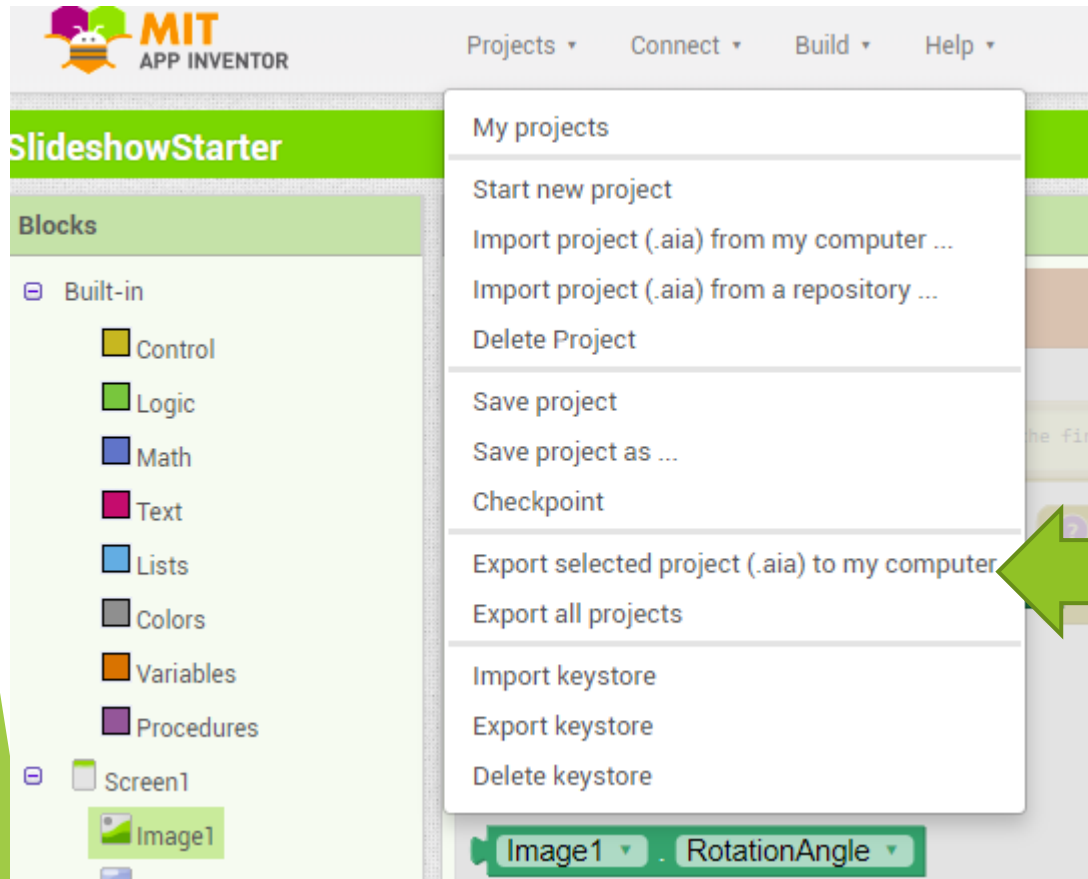
יש לבחור באופציה
טעינת הקובץ מהמחשב

קבצים של APP INVERTOR נשמרים עם סיומת

AIA

שמירת אפליקציה למחשב

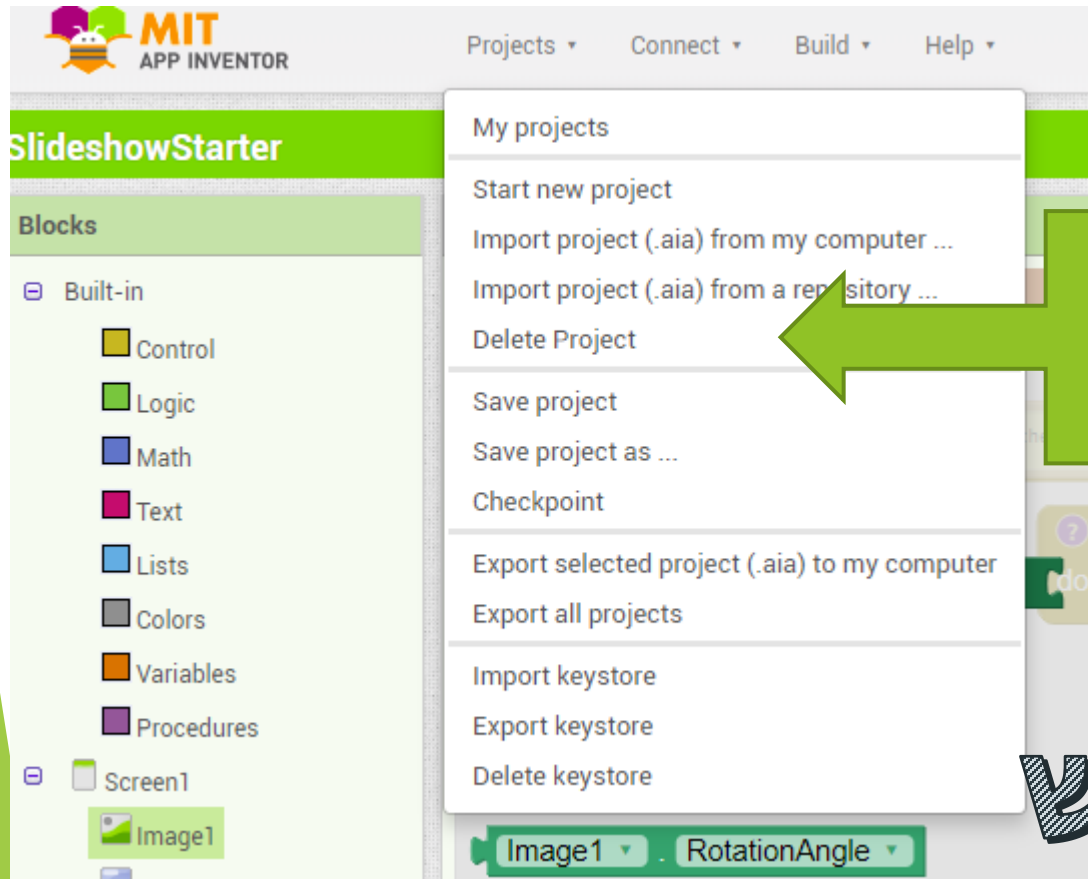
- ▶ ניתן גם לשמור את האפליקציה שפיתחנו חזרה למחשב... תחת קובץ AIA
- ▶ ניתן גם לשמור את האפליקציות ברשימת האפליקציות ב APP INVERTOR
- ▶ האפליקציות נשמרות תחת משתמש
- ▶ GMAIL שלנו



לשמור תוכנית למחשב

מחיקת אפליקציה

- ▶ חלק חשוב בעבודת התכניתן הוא ביצוע סדר לקוד
- ▶ לא תמיד כדאי לשמור את כל האפליקציות או חלקי האפליקציות שפיתחנו
- ▶ אפשר למחוק את האפליקציה מהרשימה.



מחיקת האפליקציה

סדר בקוד = סדר בראש

תרגיל מספר 3 - שמירה וטעינת אפליקציות

- ▶ בצעו מעבר על רשימה הפרויקטים שכבר הכנתם
- ▶ מחקו מיותרים
- ▶ שמרו את הפרויקט תמונות שהכנתם לקובץ AIA
- ▶ טענו את הפרויקט שוב מהקובץ

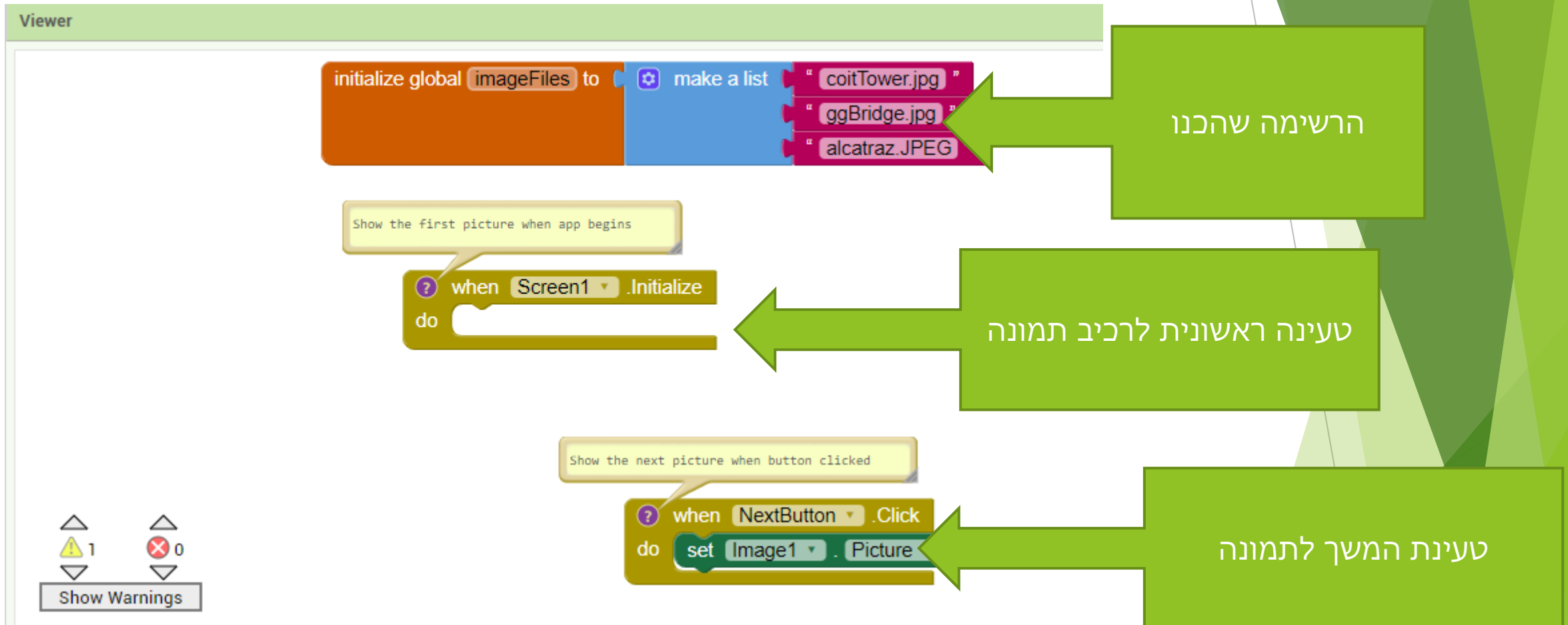
תרגיל מספר 3 - מצגת תמונות

- ▶ נניח שקיבלתם משימה להכין אפליקציה שמציגה תמונות
- ▶ מה תעשו???
- ▶ הכנו לכם אפליקציה התחלתית ועליכם להמשיך אותה
- ▶ יש לטעון את הקובץ SLIDESHOWSTARTER.AIA מהמחשב...

▶ מה זה קובץ STARTER ??

▶ קובץ STARTER הינו תחילת התרגיל - בחלק מהקבצים שנעביר לכם, נעביר לכם סטארטרים (התחלות מסוימות) ועליכם יהיה להמשיך אותם ולהפוך אותם לאפליקציות מלאות

תרגיל מספר 3 - מצגת תמונות



תרגיל מספר 3 - מצגת תמונות

- ▶ יש להשתמש בקובץ STARTER ולהכין מצגת תמונות
- ▶ יש לטעון בהתחלה את התמונה הראשונה
- ▶ ולאחר מכן להחליף תמונות

פיתוח אפליקציות

שיעור מספר 6

שיעור 6 - לולאה והמשך תמונות

- 1. לולאות
- 2. WHILE
- 3. FOR
- 4. FOREACH
- 5. TIMER
- 6. תרגול

תרגיל 1 - תרגיל חזרה קטן

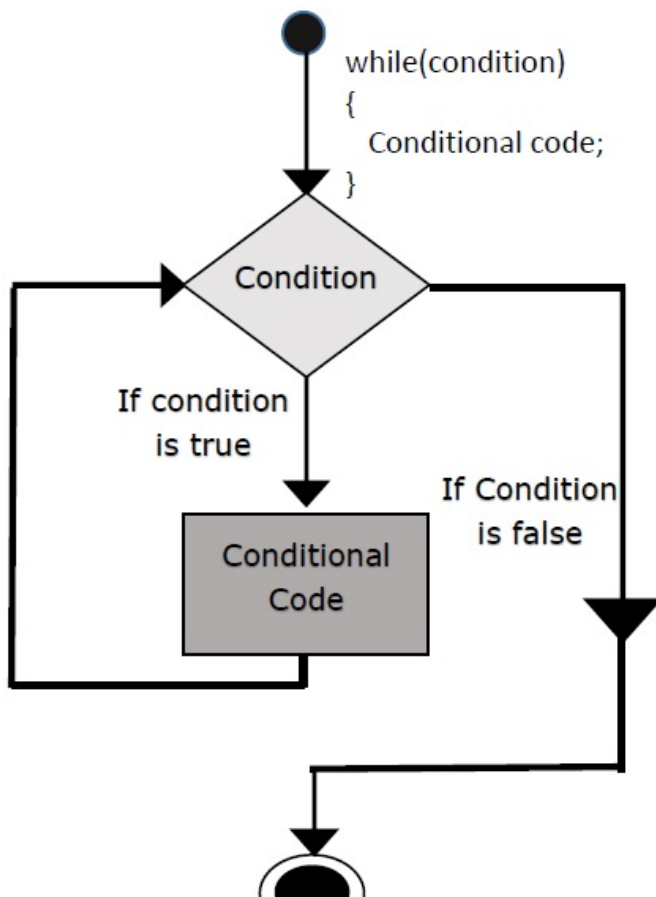
- יש לפתח אפליקציה עם תיבת טקסט ובה משתנה מסוג מחרוזת
- יש להגדיר רשימה LIST של שמות
- יש להוסיף את המשתנה לרשימה



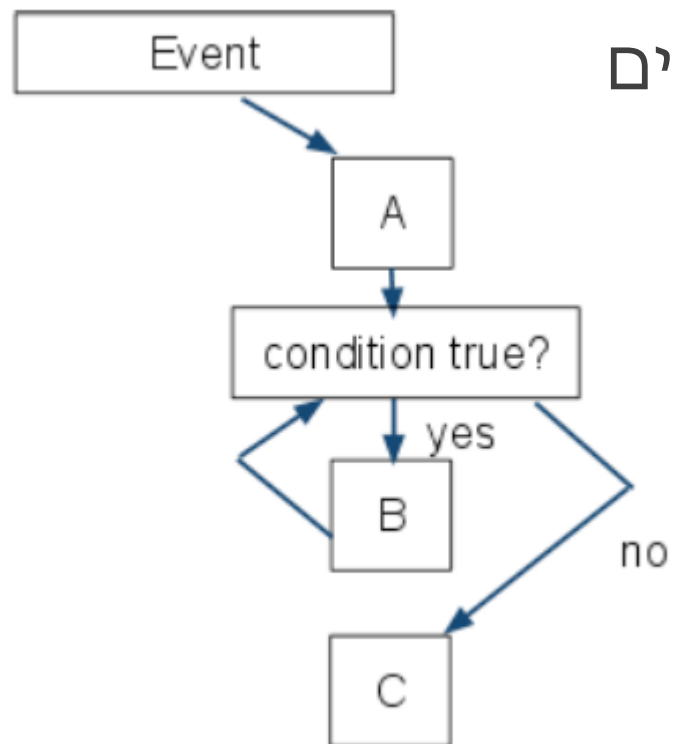
- ▶ היכולת החזקה הידוע של המחשב היא לבצע פעולות חוזרות
- ▶ מחשב, ברגע שהוא הבין מה צריך לעשות, יכול לבצע את אותה פעולה, 10, 100, ואפילו 1000 פעמים
- ▶ בדיוק לצורך כך קיימת לולאה

לולאות

לדוגמא - בתרשים כאן, מחשב כל פעם
בודק את תנאי (CONDITION) במידע
והתנאי לא מתקיים הלולאה תפסק
כל עוד תנאי מתקיים, הלולאה תמשיך
להתבצע



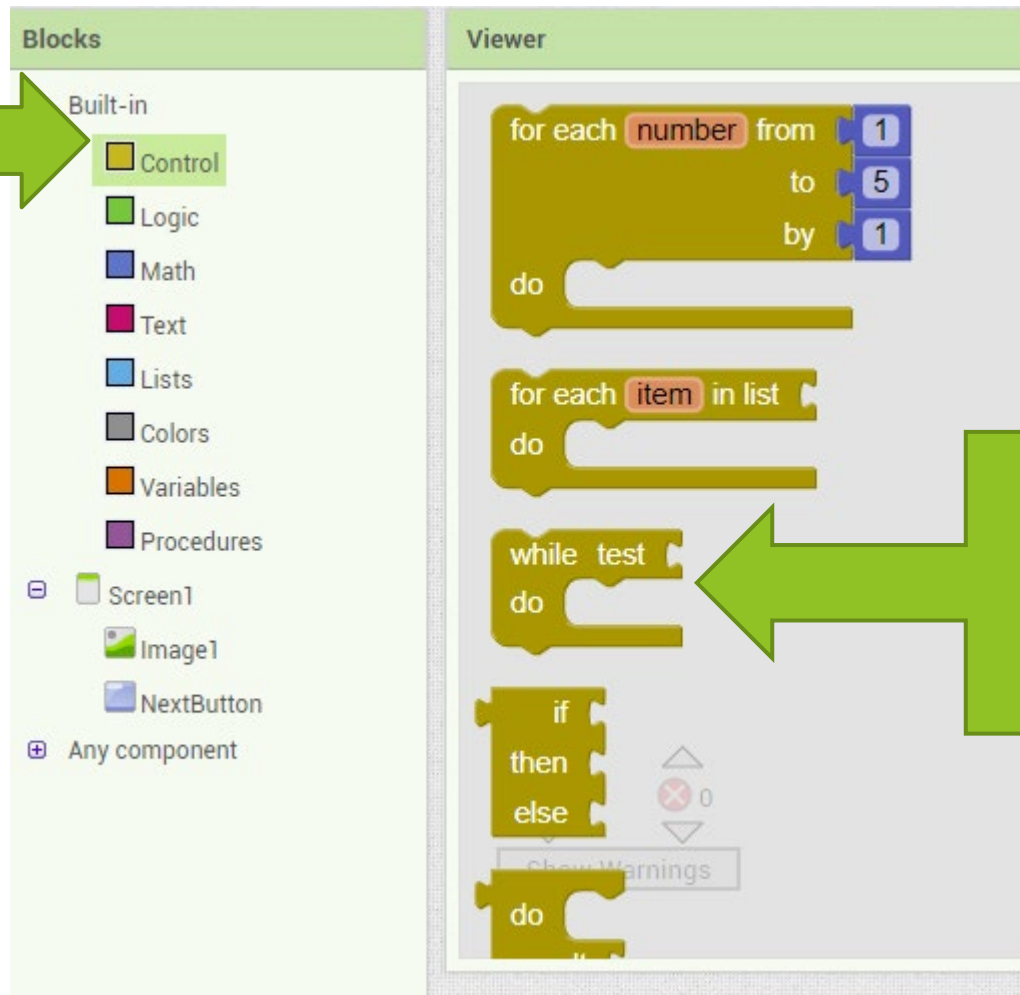
- ▶ גם בתרשים הנל התמונה דומה
- ▶ הלולאה מתבצעת כל עוד התנאי מתקיים



- ▶ ב APP INVERTOR בפרט ובתכנות בכלל קיימים סוגים שונים של פקודות המפעילות לולאות
- ▶ נלמד את חלק מהם..

לולאות

בחירת חלק
CONTROL



בחירת סוג לולאה

לולאות WHILE



- ▶ לולאת WHILE - החלק של DO מתקיימת כל עוד מתקיים התנאי בTEST, הכוונה כל עוד התנאי מחזיר TRUE
- ▶ למשל אם רוצים לבצע פעולות מ 1 עד 100
- ▶ מגדירים X1 שווה 1
- ▶ מגדירים X2 שווה ל 100
- ▶ מקדמים את X1 ב 1 על שהוא יהיה שווה ל X2



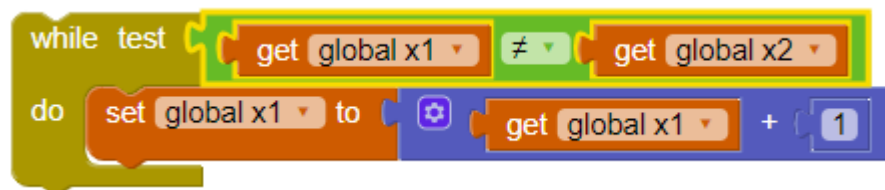
לולאות WHILE - תרגיל מספר 2



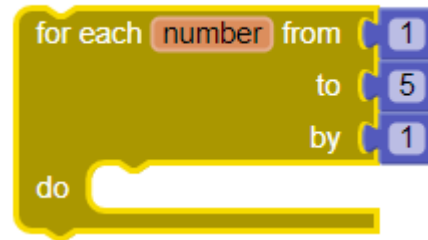
יש לפתח אפליקציה המבצעת כפל בין שני מספרים באמצעות חיבור.

כידוע אם רוצים להכפיל את 6×5 הדבר זהה לביצוע חיבור של $5+5+5+5+5+5$

יש לכתוב, באמצעות WHILE אפליקציה שמבצעת זאת

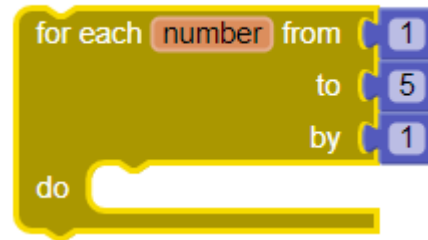


לולאות FOR



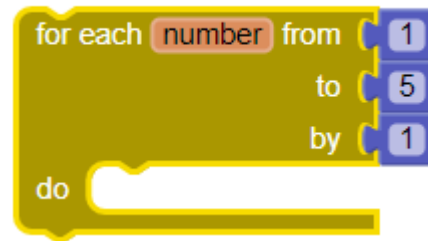
- ▶ לולאת FOR - לולאת FOR דומה ללולאת WHILE. ההבדל הוא שבמקרה של FOR אין צורך לקדם את המשתנה התנאי
- ▶ אם אנו רוצים לעשות 5 צעדים אז מגדירים את number ל-1
- ▶ לאחר מכן מקדמים את number כל פעם ב-1 עד שהוא מגיע להיות 5
- ▶ כל הפעולות המופיעות ב-DO, מתבצעות 5 פעמים

לולאות FOR



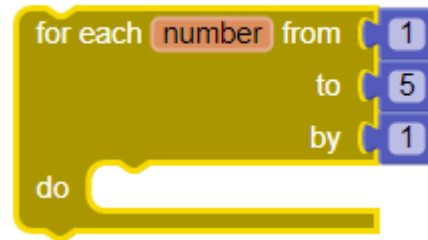
- ▶ לולאת FOR מאוד נוחה, כאשר רוצים לבצע מעבר על רשימה LIST
- ▶ ניתן להתחיל FOR לא ממספר 1
- ▶ ניתן להתקדם לא בצעדים של 1
- ▶ לולאת FOR יכולה לעזור לנו לעבוד הרבה יותר מהר

לולאות FOR



- ▶ ניתן לשלב TIMER בעבודה עם לולאת FOR
- ▶ אחרי שילוב TIMER יהיה עיקוב של כמה שניות במעבר משלב לשלב
- ▶ ביצוע שלב אחד בלולאת FOR נקרא אינטרציה INTERATION

לולאות FOR - תרגיל מספר 3



- ▶ כתוב אפליקציה אשר שקולטת שני מספרים . האפליקציה מציגה כמות * לפי מספרים שהוגדרו
- ▶ למשל אם נקלטו מספרים 1 ו 5 , אפליקציה תדפיס 5 כוכביות
- ▶ למשל אם נקלטו מספרים 1 ו 10, תציג 10 כוכביות

לולאות FOREACH



- ▶ לולאת FOREACH - לולאת FOREACH דומה קצת ללולאת FOR בכך שגם היא מתקדם את כמות הצעדים בצורה אוטומטית
- ▶ לולאת FOREACH - עוברת על כל LIST ומבצעת עבור כל איבר את פעילות

לולאות FOREACH

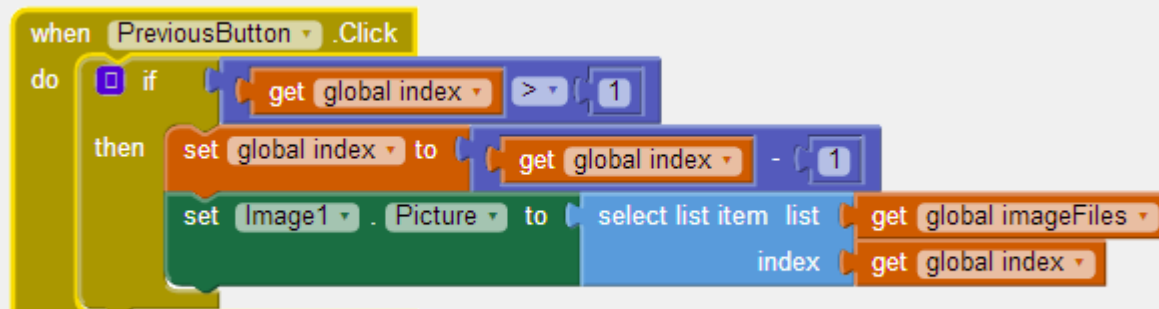


- ▶ גם בלולאת FOREACH ניתן להוסיף פעולות נוספות
- ▶ ניתן לחבר לולאות ולבצע לולאת FOREACH בתוך לולאת FOR
- ▶ חשוב אם מחברים לא לבלב את שימוש בINDEX של הלולאות

לולאות FOREACH - תרגיל מספר 4 המשך לאפליקציית תמונות



- יש לטעון את האפליקציה SLIDESHOW
- האפליקציה מכילה LIST של תמונות
- יש להוסיף לאפליקציה יכולת מעבר בין התמונות באמצעות INDEX
- יש להוסיף לאפליקציה כפתורים NEXT וPREVIOUS
- צפו בדוגמא למטה לצורך הבנת את דרך המעבר בין התמונות



TIMER שעון עצר

- ▶ תפקידו של TIMER הוא הפעלת חלק אפליקציה לאחר עבודת השעון
- ▶ מפעילים שעון כך שיקפוץ בעוד X זמן . לאחר X זמן אפשר להגדיר לאפליקציה מה לבצע
- ▶ INTERVAL - מגדיר כל כמה זמן קופץ השעון
- ▶ TIMERENABLED - קובע ממתי השעון מתחיל לפעול

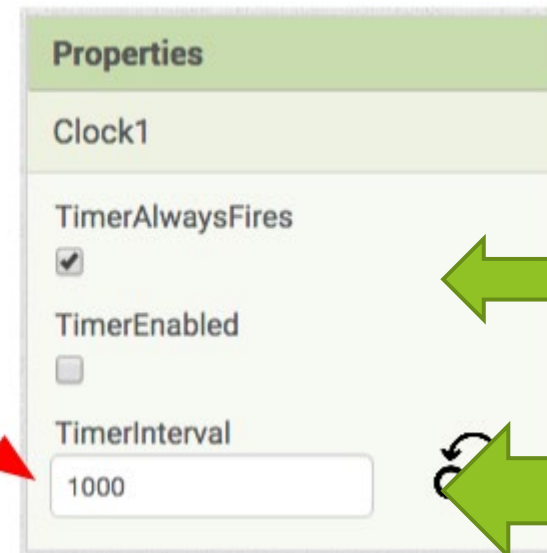
TIMER שעון מעורר

כאשר קופץ השעון, נגן
צליל SOUND

when Clock1.Timer
do call Sound1.Play

initialize global Demo to

when Clock1.Timer
do
 set Clock1.TimerEnabled to false
 while test "Still" = "What to do it"
 do set global Demo to "I do what I like"



הפעלת שעון מראש

הגדרת קפיצה שעון
כל שנייה

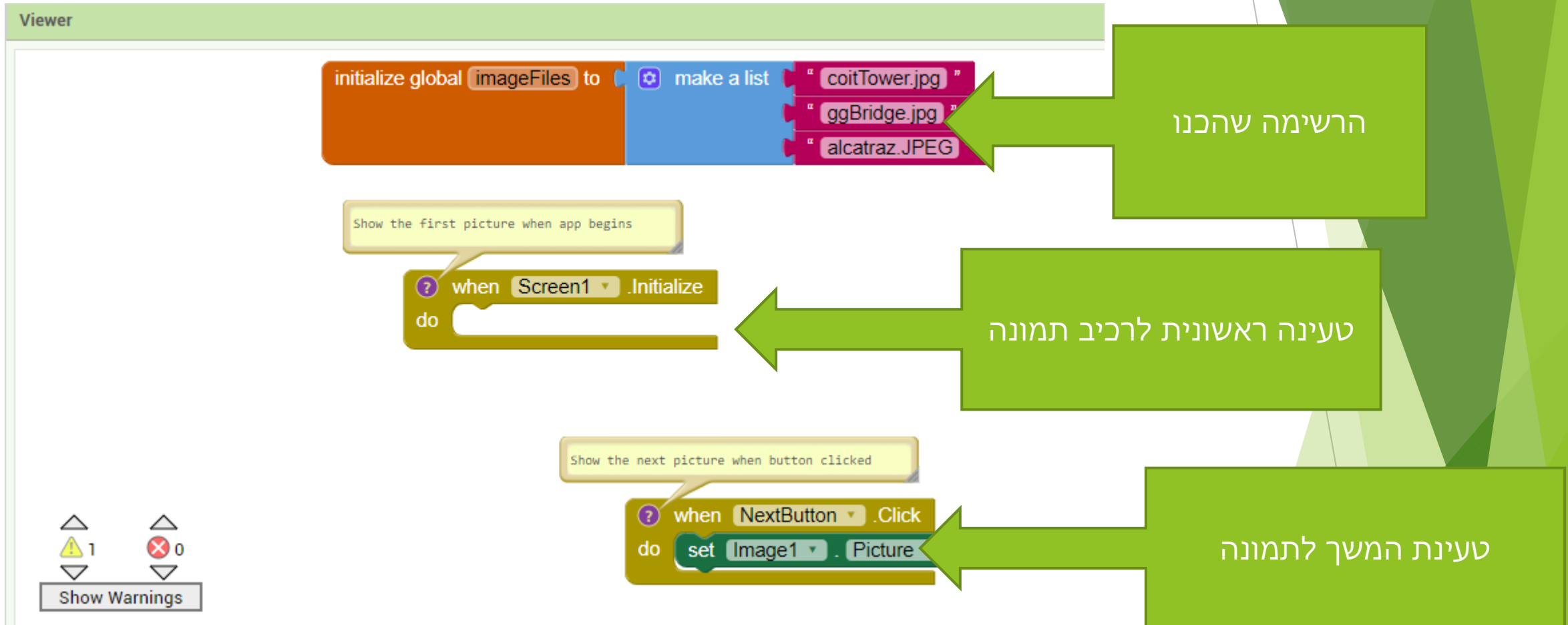
- ▶ יש לפתח אפליקציה המציגה TEXTBOX 3
- ▶ כל TEXTBOX מופיע (הופל לVISIBLE) לאחר 3 שניות אחרי הקודם
- ▶ תרגיל די פשוט ולכן מי שמסיים מוזמן לתרגיל מסכם של היום - מצגת תמונות

תרגיל מספר 6 - מצגת תמונות

- ▶ היום נפתח מצגת תמונות - ממש כמו בPOWERPOINT
- ▶ מצגת שלנו תכיל LIST של תמונות קיים מראש.
- ▶ מצגת תעבור ותציג תמונה תמונה מתוך מצגת בהפרש של כמה שניות בין תמונה לתמונה
- ▶ נשמע מורכב? אל תדאגו - נעזור לכם...

בשיעור הקודם קיבלנו וטענו קובץ STARTER והצגנו מצגת
תמונות

תרגיל מספר 6 - מצגת תמונות



תרגיל מספר 6 - מצגת תמונות

- ▶ בשיעור זה עלינו להוסיף לולאה אשר מאפשרת הצגת תמונות ב-TIMER ובלולאה
- ▶ להוסיף ריצה של 10 שניות בין הצגה להצגה

פיתוח אפליקציות

שיעור מספר 7

שיעור 7 - מספר אקראי, סנסורים וטריוויה

1. חזרה
2. RANDOM
3. סנסורים
4. סנסור ACCELERATOR
5. רשימה בתוך רשימה (LIST OF LISTS)
6. תרגיל קובייה
7. תרגיל טריוויה

תרגיל מספר 1 - תרגיל חזרה קטן

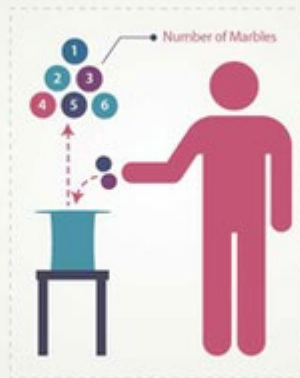
- ▶ במידה ולא סיימתם בשיעור הקודם, יש להמשיך את הפיתוח אפליקציית תמונות
- ▶ במידה וסיימתם יש להוסיף כפתור - עצור הקרנה וכפתור המשך הקרנה
- ▶ ז"א אם התמונות מוצגות ב-TIMER, ניתן יהיה ללחוץ על הכפתור ולעצור אותו

SIMPLE RANDOM SAMPLE



1st Pick

Tommy randomly picks 2 marbles from the hat containing 6 different colored marbles.



1st Pick Returned

After randomly picking 2 marbles Tommy puts the marbles back into the hat. The hat now contains 6 marbles.



2nd Pick

Tommy randomly picks 2 marbles from the hat again. He has the same chance of picking the same colored marbles as he did on the first pick.

תכנות אקראי

- ▶ באנדרואיד ניתן גם להגריל מספרים שונים.
- ▶ למשל אם אנו רוצים להגריל מספר בין 1 ל-10, ניתן להשתמש ב-MATH והוא ידע להגריל לנו מספר
- ▶ מדוע נרצה להגריל?
- ▶ למשל בזמן המשחק, אנחנו נציע לשחקן אפשרויות אקראיות מתוך מגוון האפשרויות והוא ילמד להתמודד איתם
- ▶ מובן, כי ניתן להשפיע על התוצאה ולא להשאיר את כל ליד הגורל.

תכנות אקראי RANDOM

- כך נראה תכנות RANDOM בשפת JAVA.... נראה מפחיד?

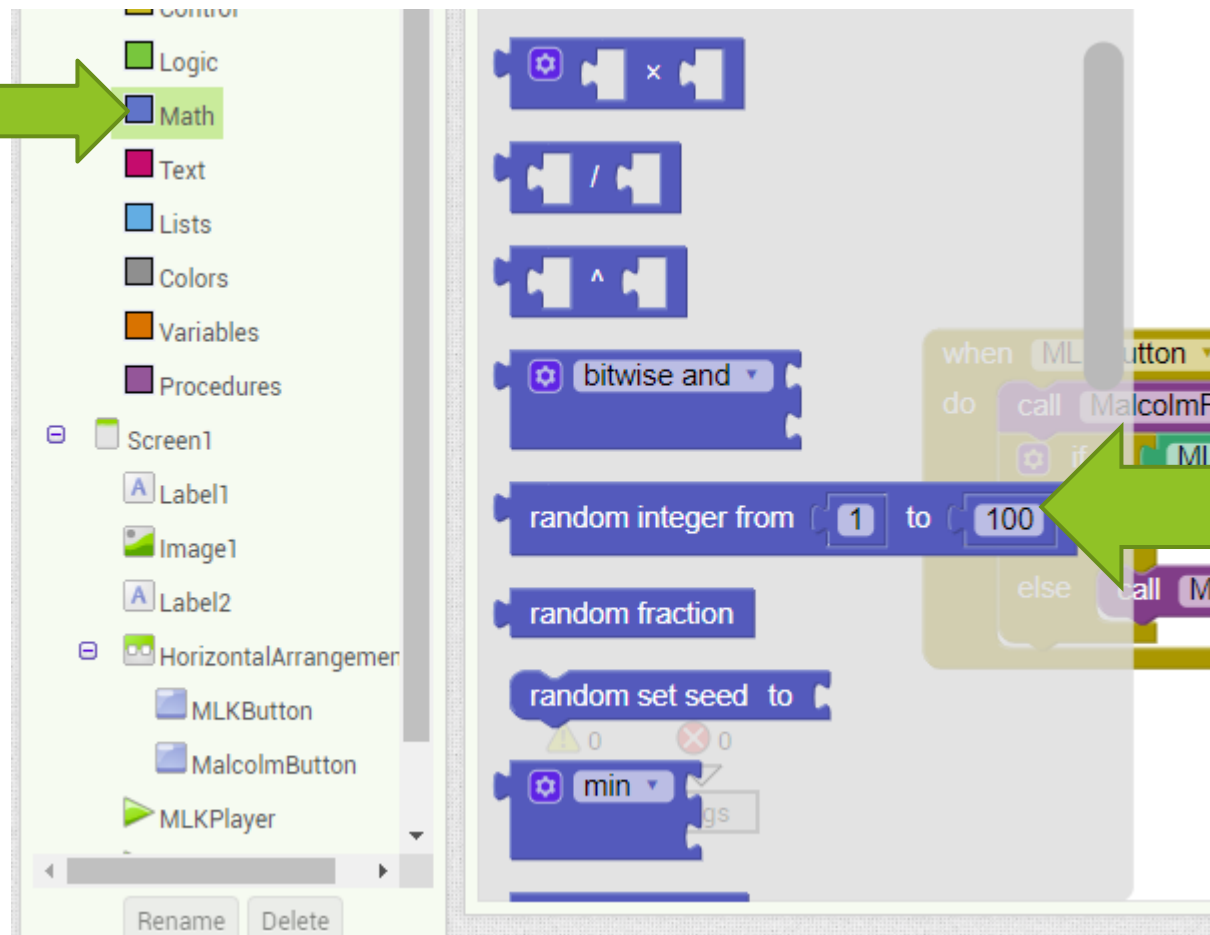
Random answer

```
// Rolls two dice until a sum of 7 is reached.
import java.util.*;

public class Dice {
    public static void main(String[] args) {
        Random rand = new Random();
        int tries = 0;
        int sum = 0;
        while (sum != 7) {
            // roll the dice once
            int roll1 = rand.nextInt(6) + 1;
            int roll2 = rand.nextInt(6) + 1;
            sum = roll1 + roll2;
            System.out.println(roll1 + " + " + roll2 + " = " + sum);
            tries++;
        }
        System.out.println("You won after " + tries + " tries!");
    }
}
```

תכנות אקראי RANDOM

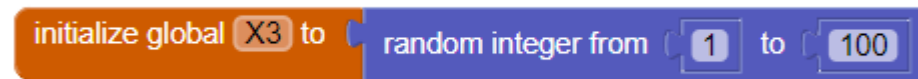
ספריית MATH



אפשרות להגדרת מספר
מ 1 ל 100

תכנות אקראי RANDOM

למשל - מגרילים מספר רנדומלי בין 1 ל-100 ומעדכנים את X3



initialize global X3 to random integer from 1 to 100

תרגיל מספר 2 - מספר אקראי

- ▶ משחק למי מוגרל מספר יותר גדול
- ▶ מוצגים 2 כפתורים
- ▶ 2 שחקנים לוחצים בו זמנית על הכפתור ומוצג מספר
- ▶ מי שמקבל מספר יותר גדול - ניצח

שימוש בסנסורים

- ▶ בפיתוח אפליקציות אנחנו יכולים להשתמש ביכולות שונות של מכשיר הסלולארי.
- ▶ אחד היכולות היא חיישנים - סנסורים
- ▶ תפקידם של הסנסורים במכשיר הסלולארי הוא לזהות תנועות, נטיות ותזוזות של המכשיר ולעדכן אפליקציות
- ▶ האפליקציות מסתנכרנות בהתאם ומציגות נתונים
- ▶ לסנסורים גם שימוש חשוב במשחקים
- ▶ שימו לב! חלק זה עדיף להריץ לא על גבי ה-EMULATOR אלא על גבי המכשיר

שימוש בסנסורים

- ▶ למעשה גם השעון וגם המצלמה הם חיישנים במכשיר
- ▶ גם הרמקול המודד צליל
- ▶ בעצם חיישנים משמשים את מכשיר סלולארי בדיוק כמו חושים משתמשים את הבן אדם
- ▶ איתם הוא שומע, מדבר, מרגיש ומתקשר עם הסביבה

סנסורים

במכשיר אנדרואיד הסטנדרטי קיימים סנסורים הבאים

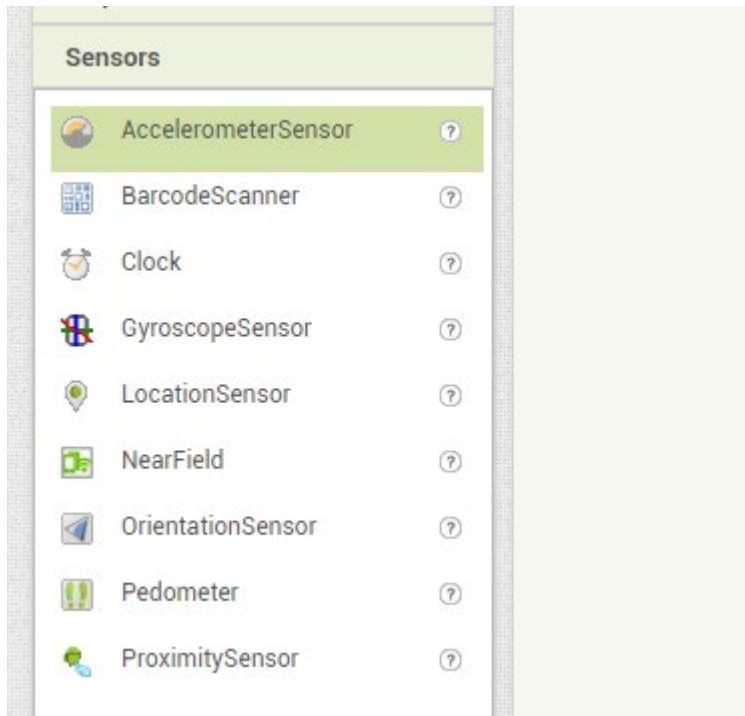
▶ סנסור (חיישן) תזוזה - Accelerometer

▶ סנסור סריקת ברקוד BARCODE SCANNER

▶ סנסור GPS והמיקום LOCATION

▶ שעון - CLOCK

(מכשיר שאחראי על מדידת זמן בתוך המכשיר)



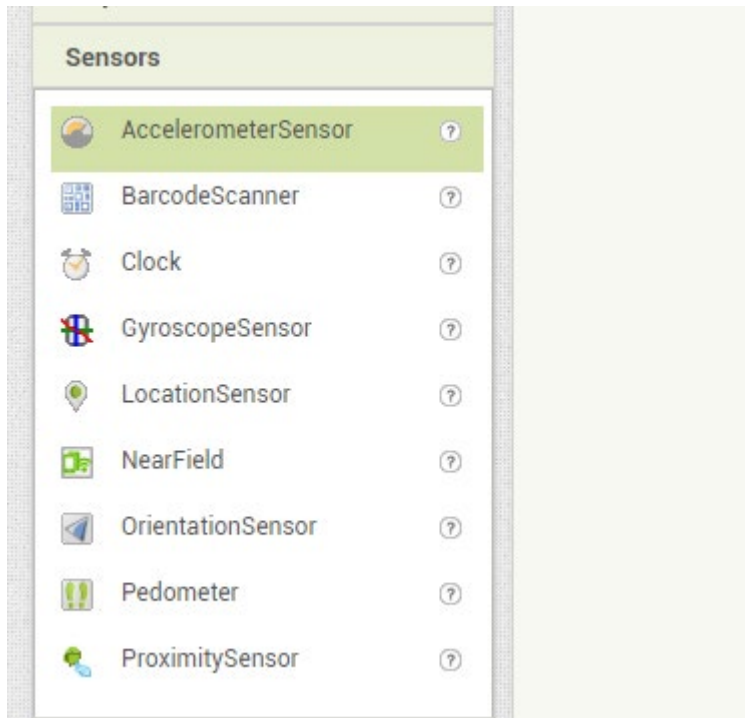
► חיישן GYROSCOPE -

האחראי על מהירות הסיבוב של המכשיר

► חיישן הכוון ORIENTATION -

חיישן המודד כוון של המכשיר מבחינת צפון ודרום

► חיישן הדיוק Proximity



סוגי סנסורים

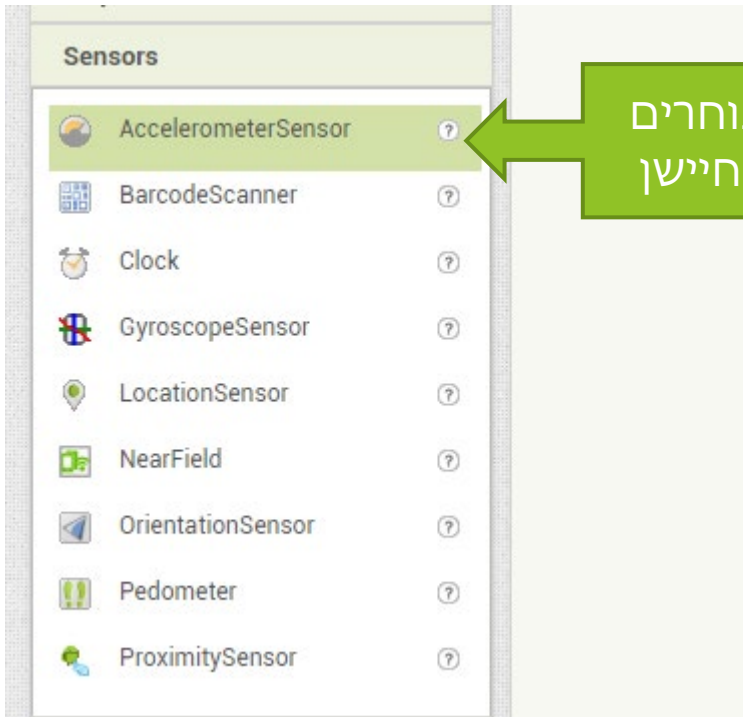
החיישנים ב-APP INVERTOR הם הרכיבים בלי נראים NON VISIBLE

כאשר נגרור את החיישן למסך הוא יתוסף למטה מתחת למסך

משתמש לא יראה שאנחנו משתמשים בחיישן

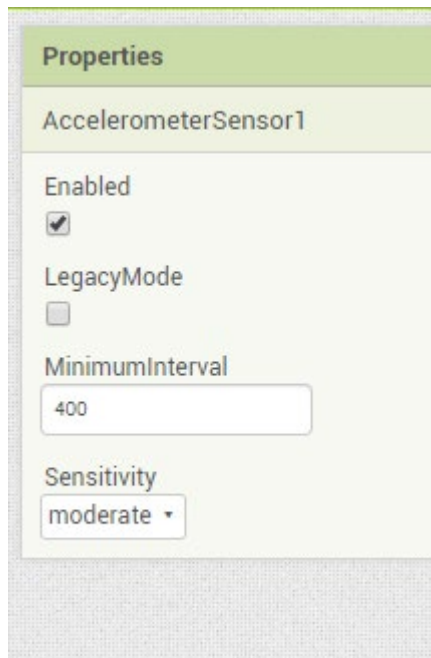
אנחנו נקבל את אירועים של החיישן

בוחרים
חיישן



גוררים את החיישן
למסך
החיישן מתוסף
מתחת למסך

חיישן תזוזה ACCELEMOMETER



- ▶ לחיישן קיימים כמה פרמטרים חשובים
- ▶ **ENABLED** - האם הוא מופעל באפליקציה או לא?
- ▶ **MINIMUMINTERVAL** - מודד מה הוא המינימום יחידות שהחיישן ירגיש שקרה משהו
- ▶ **SENSITIVITY** - רגישות - תלויה מאוד לא רק בהגדרה אלה גם באיזה סולארי יש לכם

אירועים של ACCELEROMETER

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - TextBox1
 - Button1
 - AccelerometerSensor1
- Any component

Rename Delete

Media

Upload File ...

Viewer

```
when AccelerometerSensor1 . AccelerationChanged
do
  xAccel yAccel zAccel

when AccelerometerSensor1 . Shaking
do

AccelerometerSensor1 . Available
AccelerometerSensor1 . Enabled
set AccelerometerSensor1 . Enabled to
AccelerometerSensor1 . MinimumInterval
set AccelerometerSensor1 . MinimumInterval to
AccelerometerSensor1 . Sensitivity
set AccelerometerSensor1 . Sensitivity to
AccelerometerSensor1 . XAccel
AccelerometerSensor1 . YAccel
AccelerometerSensor1 . ZAccel
AccelerometerSensor1
```

נקרא כל פעם כאשר החיישן מרגיש תנועה
החיישן מדווח וגם שולח את כרך החדש של X Y Z

SHAKING - החיישן מדווח שהמשתמש מנדנד את הסלולארי - נוח כאשר אנחנו רוצים שבמקרה של נדנד יקרה משהו

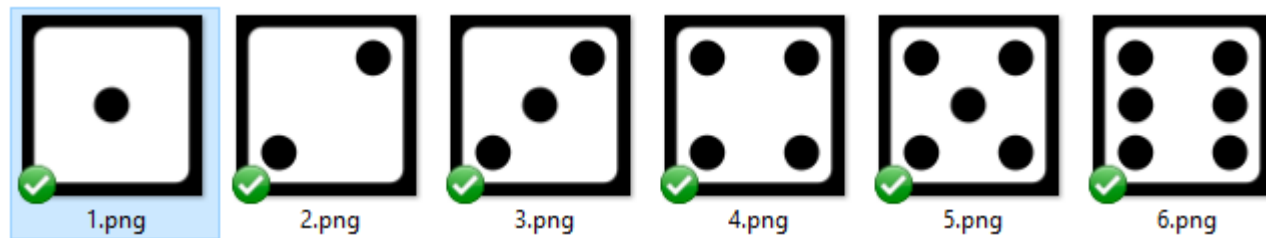


תרגיל מספר 3 - זריקת קובייה

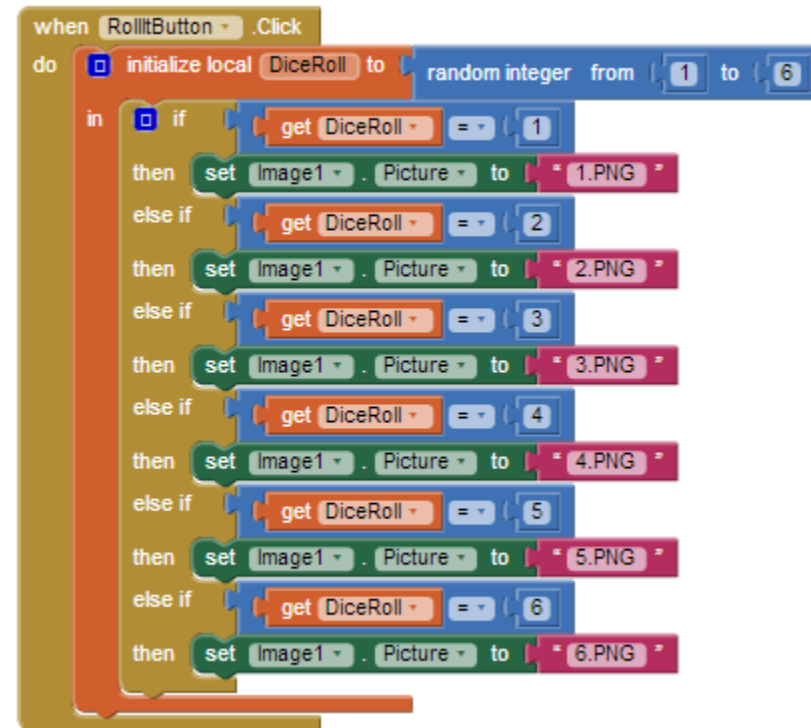
- יש לפתח אפליקציה לזריקת הקובייה DICE
- הקובייה מכילה כיודע 6 צדדים.
- מעל כל צד מופיע מספרים מ1 עד 6
- משתמש יכול להפעיל את האפליקציה באחד משני דרכים - לחיצה על כפתור למטה או נדנוד SHAKING של המכשיר

תרגיל מספר 3 - זריקת קובייה

- יש להכין מראש תמונה לכל צד של הקובייה...
- לאחר הפעלת RANDOM ניתן יהיה להציג את הקובייה המתאימה



תרגיל מספר 3 - זריקת קוביה



רשימה של רשימות LIST OF LISTS

- ▶ ב-APP INVERTOR ניתן לא רק לבנות LIST פשוט, אלא גם לבנות LIST מורכב יותר
- ▶ LIST מורכבת יותר נקרא LIST OF LISTS - רשימה של רשימות
- ▶ כל איבר ב-LIST יכול להיות LIST בפני עצמו ולהכיל מידע שונה מאיבר אחר ב-LIST



רשימה של רשימות LIST OF LISTS

- ▶ למשל איבר ראשון יכול להכיל מידע על פרות , איבר שני יכול להכין מספרים ואיבר שלישי יכול להכיל אפילו LIST ים אחרים
- ▶ אפשר לבנות LIST בתוך LIST בצורה הדרגית
- ▶ נגדיר רשימה ראשונה

```
? initialize global list1 to ? create empty list  
? initialize global list2 to ? create empty list  
? initialize global list3 to ? create empty list
```

- ▶ נוסיף איברים לLIST השני

```
? add items to list list get global list2  
item 20  
item 30  
item 10
```

רשימה של רשימות LIST OF LISTS

ניתן גם להוסיף איברים בו זמנית לכל הרשימות ►



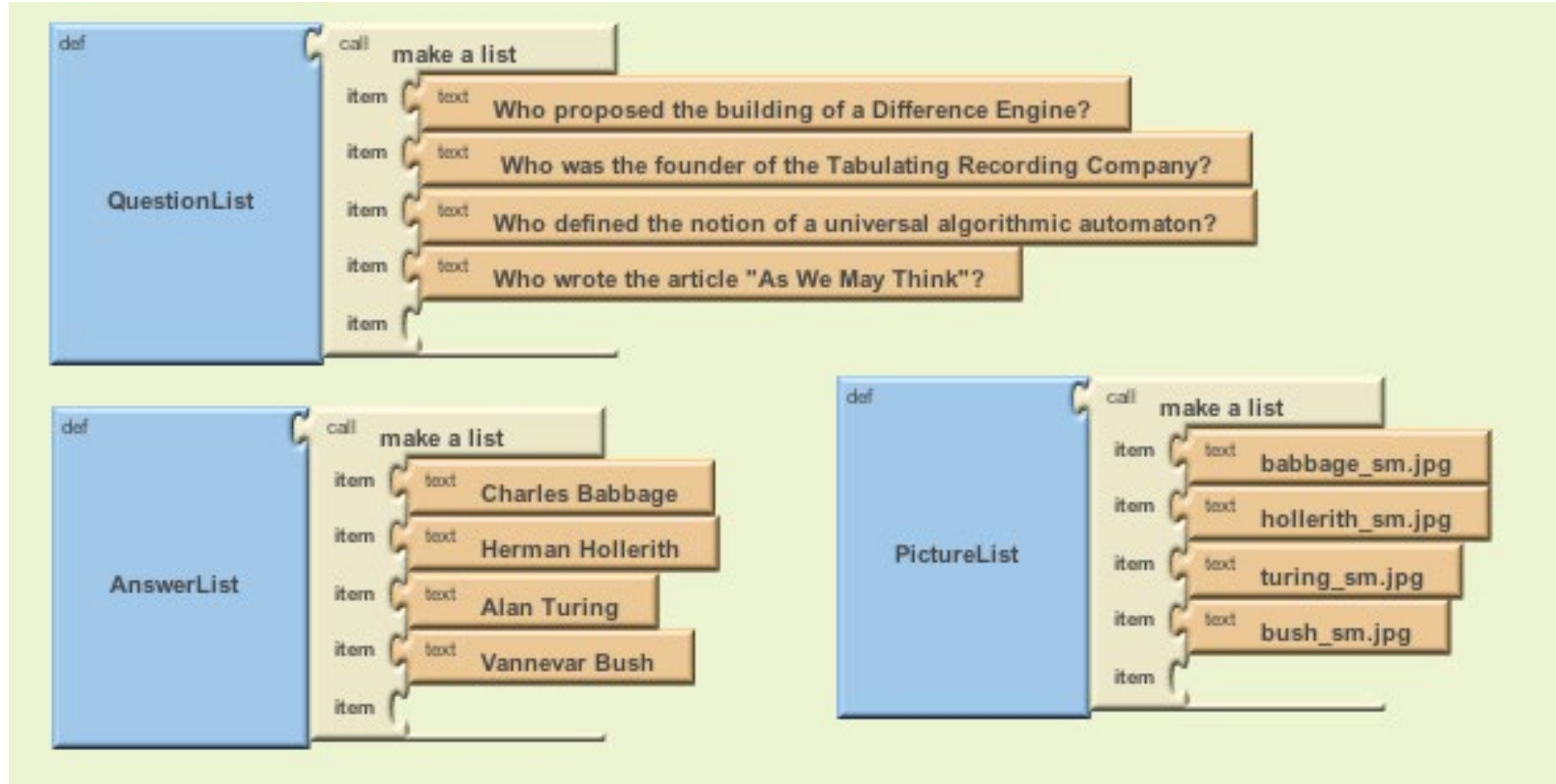
תרגיל מספר 4 - פרויקט משחק הטרוויה

- יש לפתח אפליקציה שמנהלת משחק טרוויה.
- האפליקציה מציגה לשחקן שאלות אמריקאיות
- יש להציג - <
- שאלה
- תשובה א
- תשובה ב
- תשובה ג
- תשובה ד
- יש להכין מספר שאלות מראש
- יש לבחור ולהציג שאלות בצורה אקראית

תרגיל מספר 4 - משחק הטריז

- ▶ המלצה - יש להכין 4 רשימות LISTS
- ▶ רשימת השאלות
- ▶ רשימת התשובות אפשריות - רשימה של שאלות בכל שאלה -
רשימה תשובות אפשריות
- ▶ אופציונלית - רשימת של תמונות לכל שאלה
- ▶ רשימה של תשובות נכונות
- ▶ יש להציג בו זמנית שאלה, רשימת תשובות ורשימת תמונות
- ▶ בכל תשובה יש לבדוק האם התשובה של המשתמש זהה לתשובה
נכונה ולעדכן את מספר תשובות נכונות ל + 1

תרגיל מספר 4 - משחק הטריוויה



תרגיל מספר 4 - ניתן למצוא דוגמא כאן..

<https://puravidaapps.com/quiz.php>

פיתוח אפליקציות

שיעור מספר 8

שיעור 8 - גרפיקה ו-CANVAS

1. חזרה

2. CANVAS

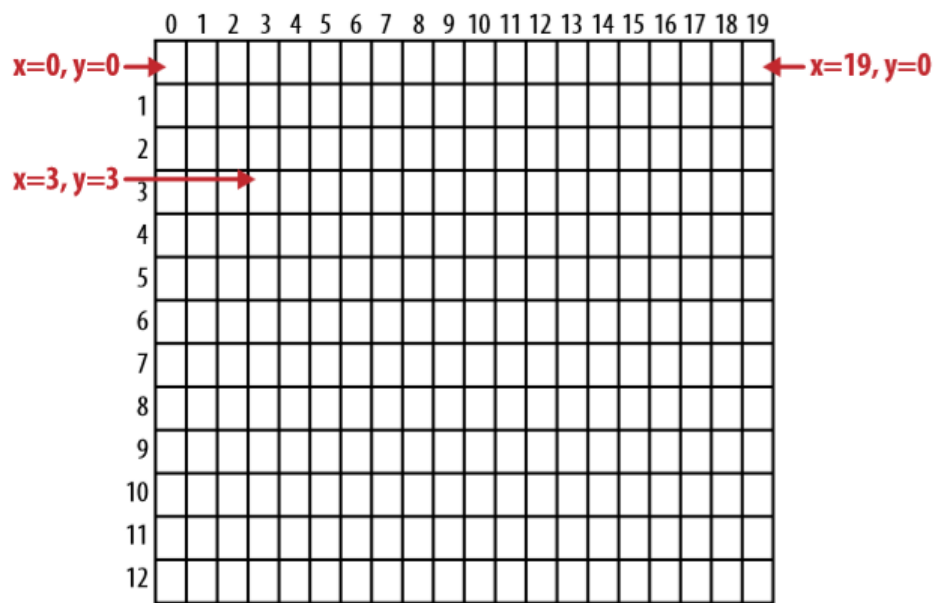
3. בשיעור הזה נתרגל המון!!

תרגיל חזרה קטן

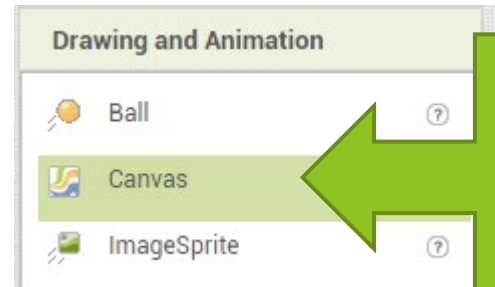
- ▶ יש לסיים את משחק הטריוויה שפיתחתם.
- ▶ במידה והמשחק הסתיים יש להוסיף תמיכה בהוספת שאלות נוספות
- ▶ תנסו לעצב יפה את צבע הרקע של השאלות והתשובות

CANVAS

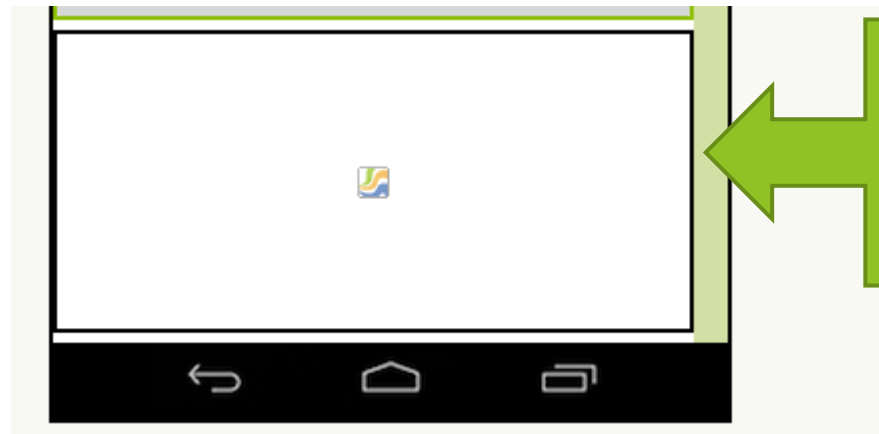
- ▶ על מנת לצייר, ולהציג ציורים וסרטים מצוירים באפליקציה, נדרשות יכולות גרפיות
- ▶ באנדרואיד, יכולות גרפיות ניתן להשיג באמצעות רכיב ה-CANVAS
- ▶ CANVAS הוא רכיב וניתן למקם אותו בכל מקום במסך.
- ▶ הרעיון הוא לחלק את מיקום הציור למערכת קואורדינטות בעלות כתובות X ו Y
- ▶ המיקום בפינה שמאלית העליונה מוגדר כ-0 0
- ▶ ככל שזזים ימינה X גודל וככל שזזים למטה Y גודל



CANVAS



בחירת
CANVAS
תחת
DRAWING



מיקום
CANVAS
על גבי מסך



מומלץ להרחיב
את CANVAS
כך שיתפרס על
כל העמוד או
לפחות שיתפרס
לרוחב

CANVAS

- ▶ הציור ב-CANVAS מתבצע בדומה לציור רגיל
- ▶ ניתן לצייר
 - ▶ נקודות POINTS
 - ▶ קווים LINES
 - ▶ עיגולים CIRCLES
 - ▶ ניתן לצייר צורות בעלות TEXT
 - ▶ ניתן לצייר קשתות ARCS
 - ▶ ניתן לצייר צורות שונות
- ▶ על מנת לצייר צורה, יש להעביר ל-CANVAS את נ"צ - נקודות ציון - כאורדינטות X וY עבור כל נקודה בציור

CANVAS

```
call Canvas1 .DrawLine  
x1  
y1  
x2  
y2
```

ציור קו -
מעבירים X וY
עבור 2 נקודות
הסוף וההתחלה

```
call Canvas1 .DrawPoint  
x  
y
```

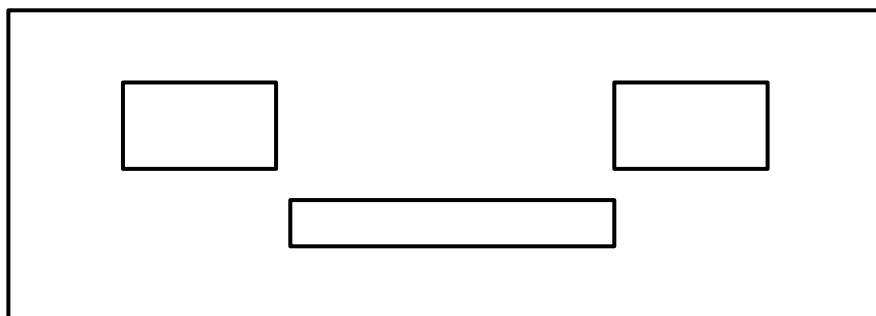
ציור של נקודה
מעבירים נ"צ
של הנקודה

```
call Canvas1 .DrawCircle  
centerX  
centerY  
radius  
fill true
```

ציור עיגול
יש להעביר X וY
ורדיוס
כמו כן ניתן לציין
האם למלא את
העיגול בצבע

תרגיל מספר 2

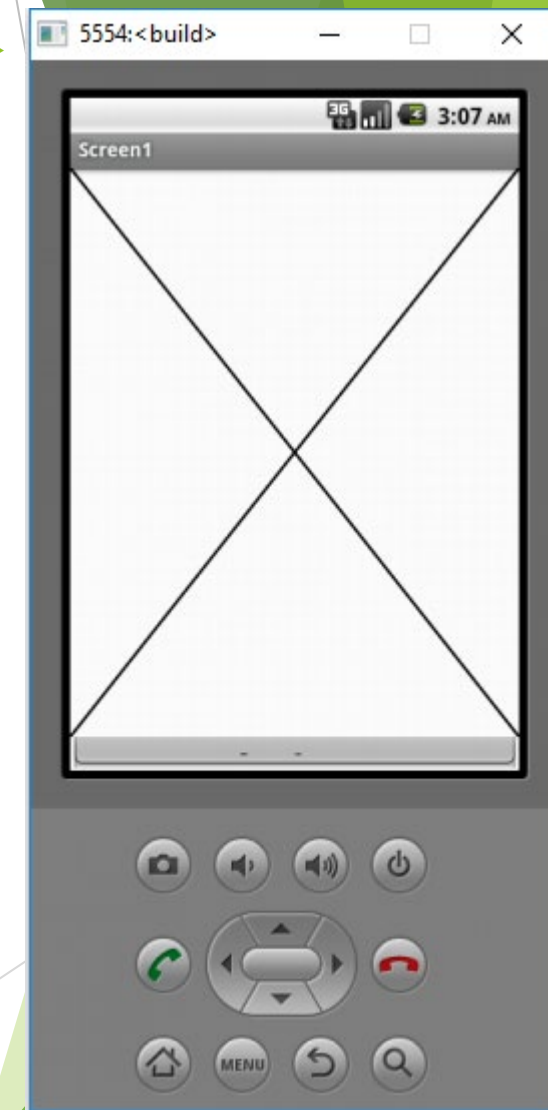
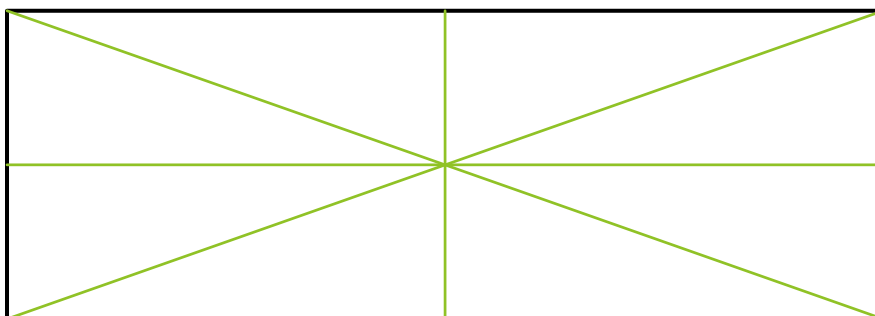
- יש לפתח אפליקציה המציגה רובוט הכי פשוט
- שימו לב, יש לצייר את רובוט באמצעות קווים ב בלבד ללא שימוש בSHAPE!!!
- רמז - הקו הבא מתחיל בדיוק בנקודה שבה הקו הקודם נגמר.
- תתחיל בערך מנקודה 1,1 או 2,2 על מנת לנצל את כל משטח



תרגיל מספר 3

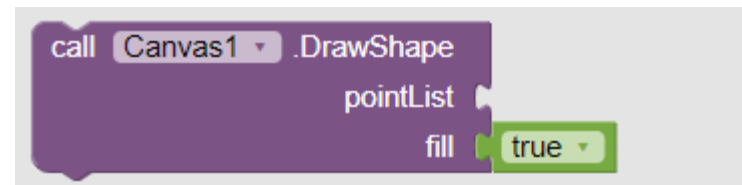
יש לצייר מספר קווים חשובים

- ▶ אלכסון ראשי
- ▶ אלכסון משני
- ▶ קו אמצע אנוכי
- ▶ קו אמצע אופקי
- ▶ שימו לב - הקווים צריכים להיפגש במרכז המסך



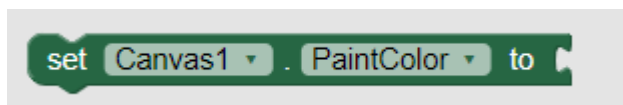
CANVAS

- ▶ ציור גרפי מאפשר לצייר כל צורה על ידי ציור פוליגון
- ▶ צורה מקבלת כל רשימה של נקודות.
- ▶ כל נקודה בעצם מסמלת פנייה
- ▶ לכן יש להעביר לPOINTLIST רשימת נקודות של הצורה
- ▶ ניתן לתכנן את ציור מראש - תתכננו את הנקודות בהם אתם תרצו לפנות

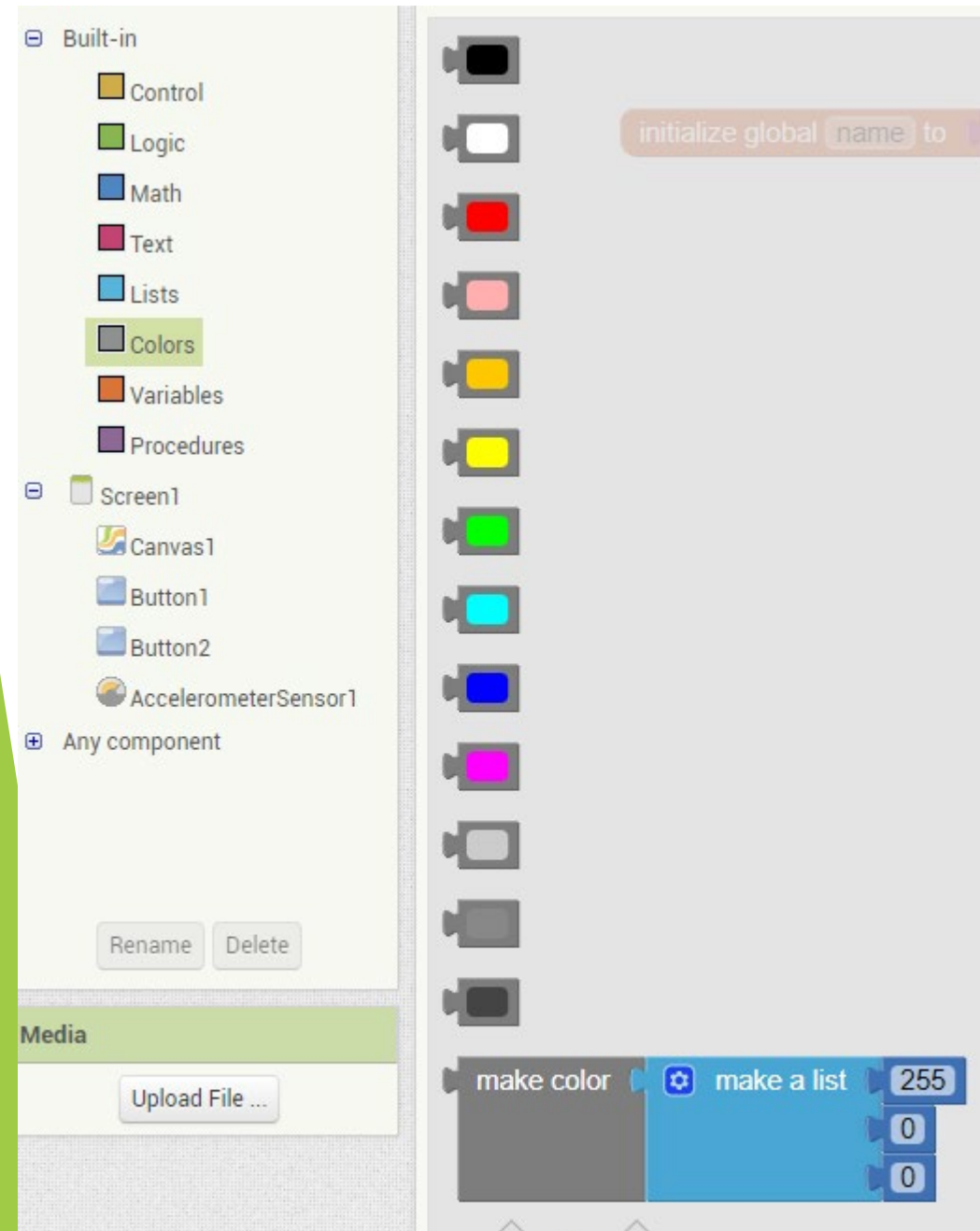


CANVAS

- ▶ CANVAS מאפשר ציור בצבעים שונים
- ▶ ציור ב-CANVAS דומה מאוד לציור בעט או בטוש רגילים
- ▶ על מנת לצייר, למשל, בצבע אדום, אנו לוקחים טוש אדום ומציירים
- ▶ עכשיו על מנת לצייר קו בצבע ירוק, אנו מחליפים טוש לירוק
- ▶ על מנת לשנות את צבע, יש לבחור צבע מאפשרויות הצבעים - < נראה זאת בשקף הבא



CANVAS

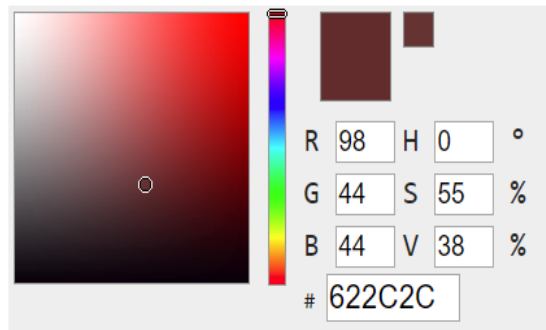


- ▶ ב APP INVERTOR ניתן לבחור צבעים קבועים או ניתן ליצור צבע חדש
- ▶ על מנת ליצור צבע חדש יש להקיש את מספרו תוך שימוש ב MAKE COLOR
- ▶ את בניית צבע ניתן לעשות דרך האינטרנט

RGB Color Codes Chart

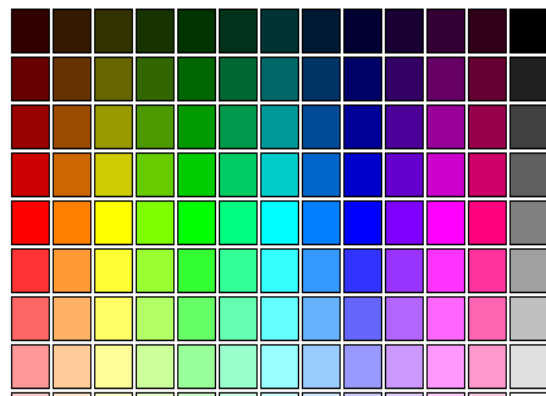
[RGB color picker](#) | [RGB color codes chart](#) | [RGB color space](#) | [RGB color format and calculation](#) | [RGB color table](#)

RGB color picker



RGB color codes chart

Hover with cursor on color to get the hex and decimal color codes below:



CANVAS עובד עם הצבעים בפורמט RGB

RGB מסמל חיבור של 3 צבעים

ADOM - RED

YROK - GREEN

KHOL - BLUE

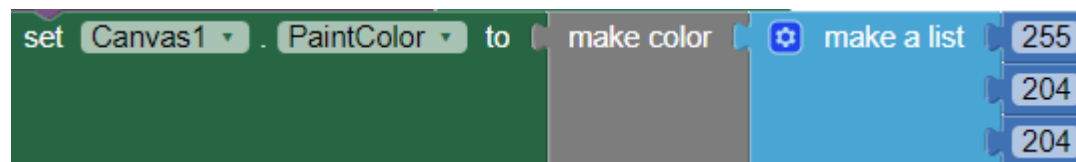
למשל באתר הזה ניתן לבנות עם הצבעים הללו כל צבע שתמצאו

https://www.rapidtables.com/web/color/RGB_Color.html

אתר מאפשר בניית צבע על ידי בחירה

אתם יכולים לבחור צבע כלשהו ואז להעתיק את מספרים R G B ל

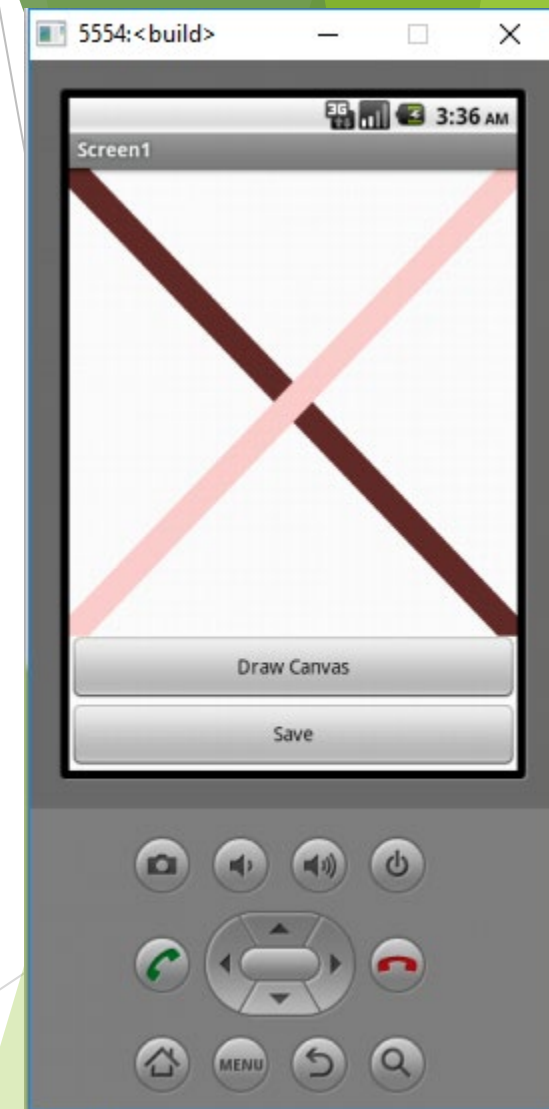
תוך MAKE A LIST



CANVAS

- ▶ בצורה דומה לשינוי צבע, ניתן גם להשפיע על עובי הקו
- ▶ עובי זהה לכמות הקווים דקים מציירים יחד
- ▶ למשל, אם כתבתם שעובי הקו זהה ל-20 - כאילו ציירתם ישר 20 קווים!!!!

```
set Canvas1 . LineWidth to 20
```



תרגיל מספר 4

יש לצייר מגן דוד ▶

מגן דוד מורכב מ 2 משולשים ▶

יש לצייר אותו בצבע כחול ▶

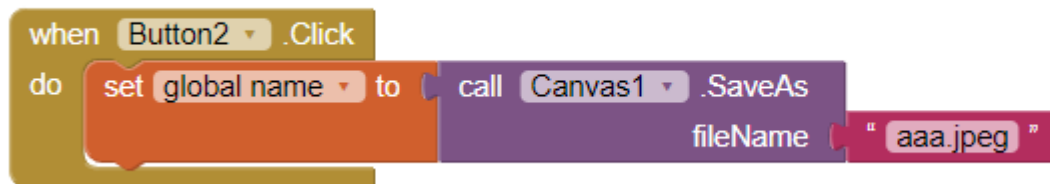
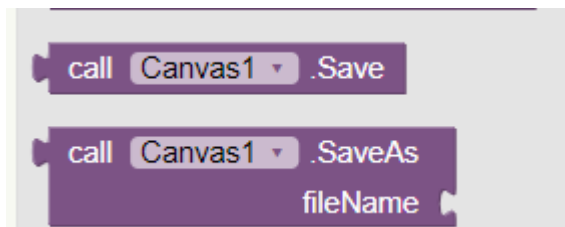
יש לקבוע את גודל הקו בהתאם ▶

ניתן להשתמש ב-SHAPE ▶



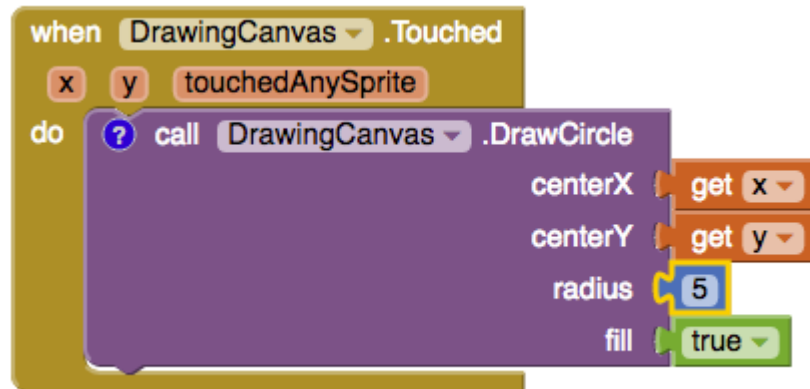
CANVAS

- ▶ CANVAS מאפשר שמירת ציור
- ▶ השמירה מתבצעת בתוך ה EMULATOR !!!
- ▶ על מנת לשמור יש גם להגדיר משתנה נוסף - משתנה המכיל את התוצאה של השמירה



אירועים CANVAS

- ▶ ניתן לצייר גם לא רק במקומות בהם תכננו אלה גם במקומות, בהם נגע המשתמש
- ▶ לצורך כך ניתן להשתמש באירועים שמציע CANVAS
- ▶ האירוע הראשון זה TOUCHED - נגע (שימו לב זה בלשון עבר!!!)
- ▶ בכל מקום בו המשתמש נוגע, ניתן לצייר ציור
- ▶ האירוע, מעביר לכם את X ואת Y שבו המשתמש נגע



תרגיל מספר 5 - ציור עיגולים

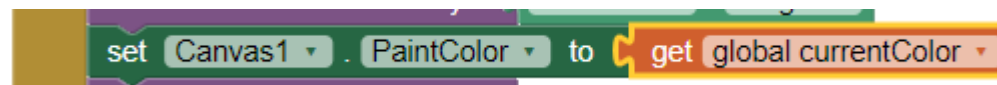
- יש לפתח אפליקציה המציירת עיגולים במקום בו משתמש מבצע לחיצה
- יש לצייר עיגול בן 10 פיקסלים

CANVAS TOUCHED אירוע

- ▶ נניח שנרצה לאפשר למשתמש שלנו לקבוע לא רק מיקום בו הוא יצייר עיגול או ריבוע
- ▶ נניח שהמשתמש ירצה גם לקבוע גם את הצורה שאותה יצייר...
- ▶ או את הצבע הציור
- ▶ ניתן להשתמש במשתנים
- ▶ נגדיר משתנה גלובאלי CURRENTCOLOR



- ▶ מעכשיו לפני שמציירים צורה מעדכנים את הצבע שלה



- ▶ עכשיו ננהל 3 כפתורים - כל כפתור במקום טקסט נצבע בצבע שונה
- ▶ אם המשתמש לוחץ על אדום, נשנה את CURRENTCOLOR לאדום
- ▶ וכך הלאה

תרגיל מספר 6 - ציור עיגולים בצבעים שונים

יש להוסיף לאפליקציה שפיתחתם בתרגיל 5, אפשרות להחליף צבעים של העיגולים

תרגיל מספר 7 - החלפת צורה לפני ציור

- ▶ עכשיו ננסה גם להחליף סוג צורה - הוסיפו באפליקציה 5 ו 6 כפתור נוסף שקובע האם לצייר ריבוע או עיגול.
- ▶ תגדירו משתנה גלובלי SHAPETYPE ששווה למחרוזת "CIRCLE"
- ▶ תוסיפו 2 כפתורים - "ריבוע" ו"עיגול" אם משתמש לוחץ על ריבוע, SHAPETYPE יהיה שווה ל"Rectangle" ואם לוחץ על עיגול, המשתמש יחזור להיות שווה ל"CIRCLE"
- ▶ עכשיו, לפני כל ציור, תוסיפו תנאי IF ו תבדקו למה שווה המשתנה SHAPETYPE, אם ל "CIRCLE", יש לצייר עיגול ואם ל"RECTANGLE", יש לצייר ריבוע

פרויקט ציור חופשי - PAINTPOT - שלב א

- ▶ עכשיו בואו נפשיל שרוולים ונצייר משהו מורכב יותר
- ▶ נעבוד על האפליקציה PAINTPOT
- ▶ המטרה היא לאפשר ציור מעל התמונה של החתול.
- ▶ המשתמש שלנו יוכל לצייר כל מה שירצה
- ▶ הוא יוכל להחליף צבעים
- ▶ בכל לחיצה הוא יכול לבחור בצבע אחר
- ▶ בשיעור הזה נכין את הכל ובשיעור נמשיך עם הציור
- ▶ יש להכין את הכפתורים ואת התמונה של החתול
- ▶ בנוסף לאפשר למשתמש לצייר עיגולים מעל התמונה
- ▶ (רמז -האפליקציה כבר קיימת ואפשר להסתכל איך עושים זאת 😊)



פיתוח אפליקציות

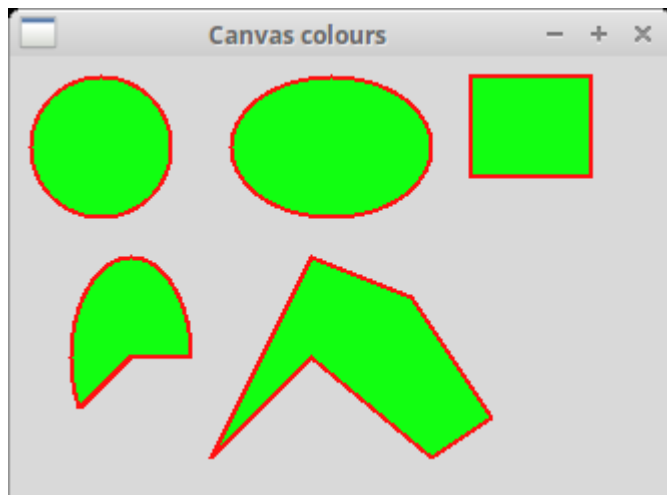
שיעור מספר 9

שיעור 9 - המשך גרפיקה ו-CANVAS

1. חזרה
2. הזזה CANVAS
3. עבודה עם SPRITE
4. ציור מתקדם ולוגיקה

תרגיל מספר 1 - תרגיל חזרה קטן

על מנת להיזכר במה שלמדנו, יש לצייר מספר צורות



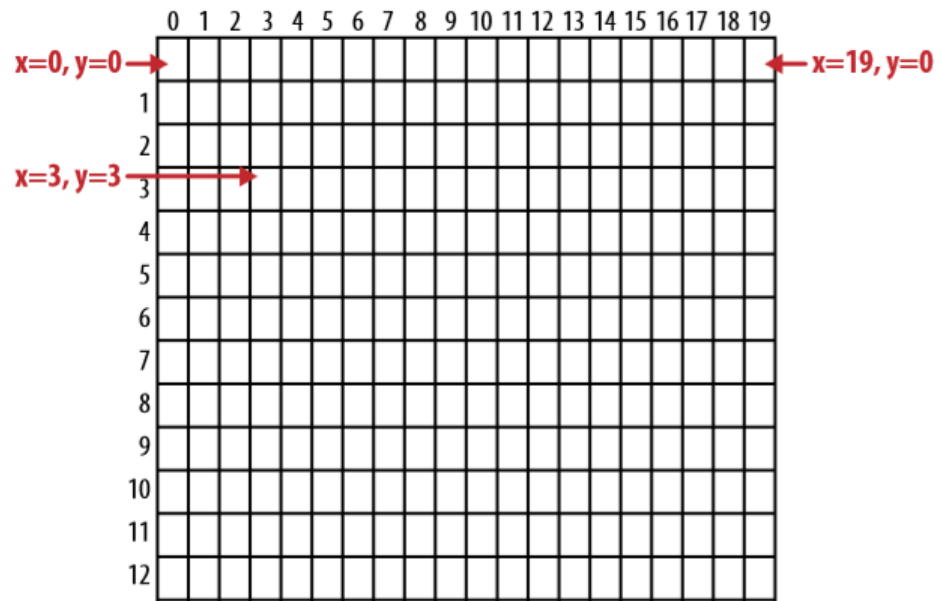
תרגיל מספר 2 - פרויקט ציור חופשי - PAINTPOT - שלב ב

- ▶ בשיעור הקודם, התחלנו לפתח את PAINTPOT
- ▶ יש בדוק שהפרויקט מוכן
- ▶ בשלב זה, אמור להופיע ציור החתול ו3 כפתורים
- ▶ בנוסף, ניתן לצייר עיגולים בצבעים שונים כפי שלמדנו.
- ▶ במידה ועוד לא ביצעת, יש להשלים את החלק הזה



CANVAS

- ▶ גם בשיעור הזה, אנו נמשיך להכיר את CANVAS
- ▶ אנו נמשיך גם לצייר צורות שונות וגם נלמד דברים מתקדמים יותר ולוגיקה



פרויקט ציור חופשי - PAINTPOT - שלב ב

עכשיו נכיר שלבים הבאים - נלמד לאפשר למשתמש לצייר כל קו שירצה ואפילו לכתוב מילים שלמות



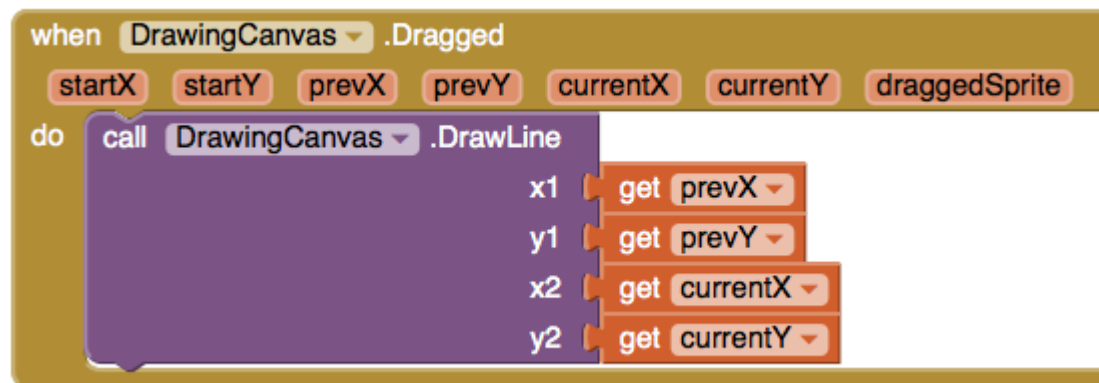
CANVAS - DRAGGED

- ▶ האירוע DRAGGED של CANVAS מאפשר לנו לבצע פעולות בזמן המשתמש מזיז כפתור מעל CANVAS
- ▶ האירוע, שולח לנו מספר פרמטרים
- ▶ startX, startY - כפתורים של X וY בשניה המשתמש התחיל בפעולה
- ▶ prevX, prevY - הX וY הקודמים (ממש שהיו בשניה הקודמת לפני המיקום הנוכחי)
- ▶ currentX, currentY - הX וY הנוכחיים.

```
when Canvas1 .Dragged
  startX startY prevX prevY currentX currentY draggedAnySprite
do
```

CANVAS - DRAGGED

- ▶ על מנת לצייר קו רציף, אנחנו צריכים כל הזמן, בכל שניה שבה משתמש מזיז אצבע מעל המסך, לצייר קו.
- ▶ בעצם אנחנו כל הזמן מחברים קו מהנגיעה הקודמת שלו למיקום הנוכחי שלנו



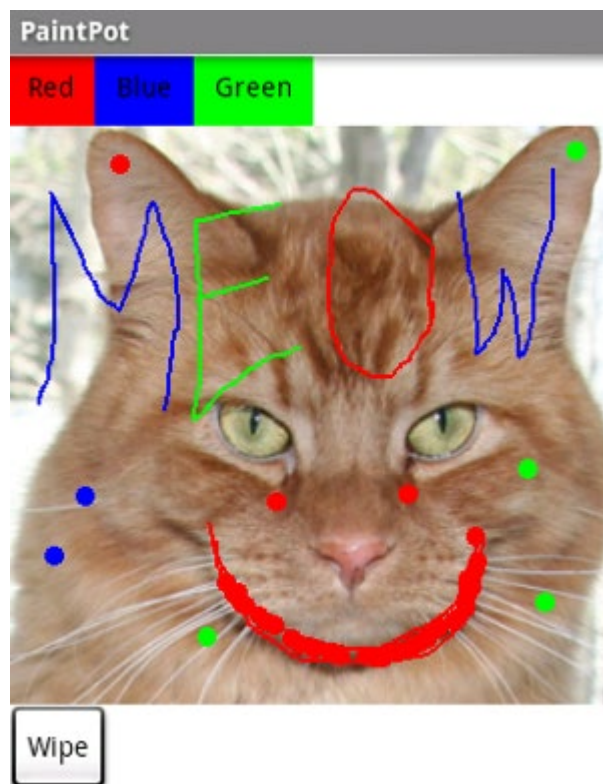
פרויקט ציור חופשי - PAINTPOT - שלב ב

- ▶ עכשיו תוסיפו את היכולת ציור מעל החתול.
- ▶ יש להוסיף גם תמיכה בהחלפת צבע בזמן הציור



פרויקט ציור חופשי - PAINTPOT - פתרון

- עכשיו תוסיפו את היכולת ציור מעל החתול.
- יש להוסיף גם תמיכה בהחלפת צבע בזמן הציור



```
when RedButton.Click
do set DrawingCanvas.PaintColor to Red

when BlueButton.Click
do set DrawingCanvas.PaintColor to Blue

when GreenButton.Click
do set DrawingCanvas.PaintColor to Green

when ButtonWipe.Click
do call DrawingCanvas.Clear

initialize global small to 2
initialize global big to 8
initialize global dotsize to 2

when ButtonBig.Click
do set global dotsize to get global big
```

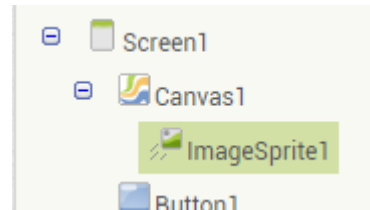
```
when DrawingCanvas.Touched
x y touchedAnySprite
do call DrawingCanvas.DrawCircle
    centerX get x
    centerY get y
    radius get global dotsize
    fill true

when DrawingCanvas.Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do call DrawingCanvas.DrawLine
    x1 get prevX
    y1 get prevY
    x2 get currentX
    y2 get currentY

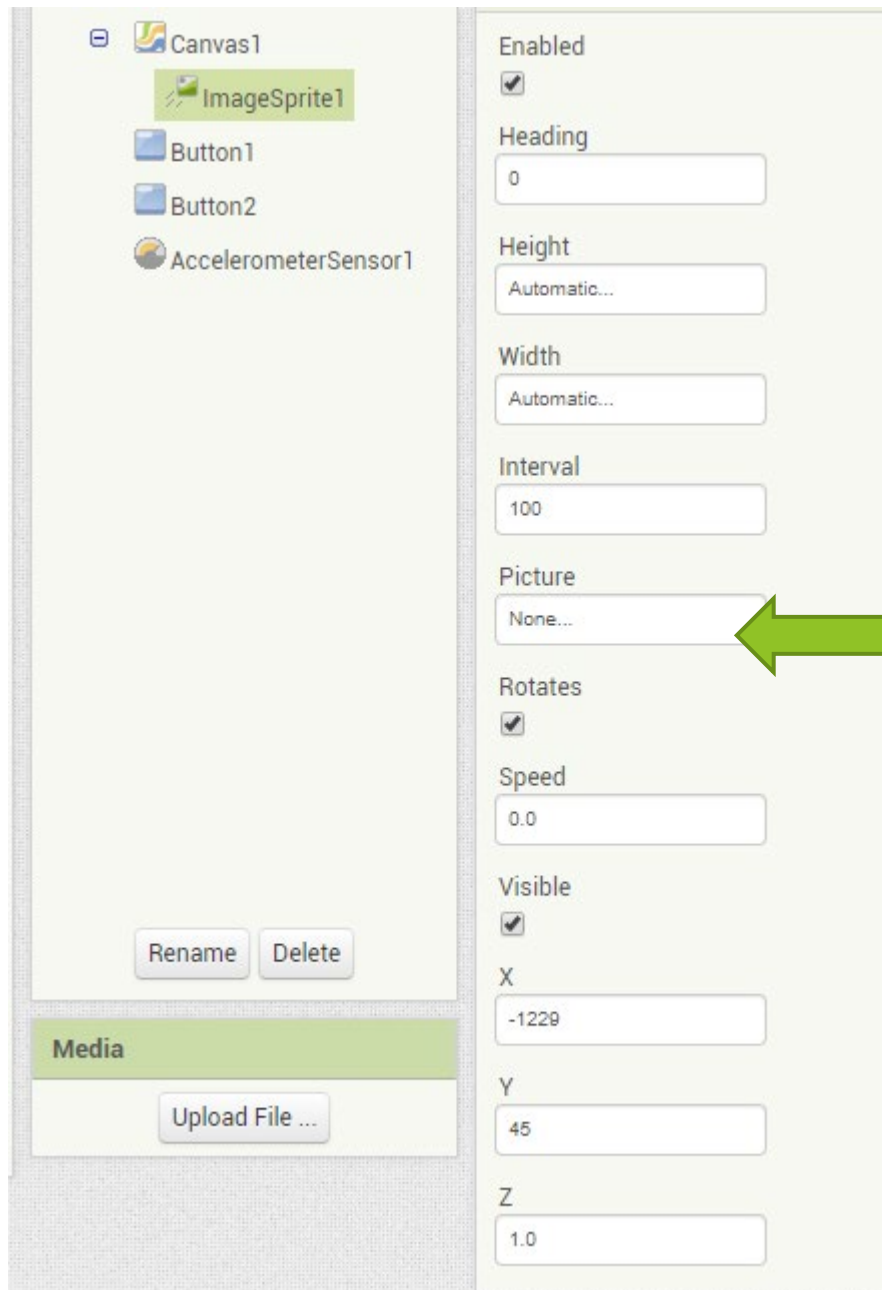
when ButtonSmall.Click
do set global dotsize to get global small
```


SPRITE - CANVAS

- ▶ CANVAS מאפשר לנו להזיז צורה קטנה (למשל תמונה) מעל גוף המצייר.
- ▶ התמונה הזאת נקראת SPRITE - SPRITE זה בעצם תמונה מעל תמונה.... (כי CANVAS הוא גם תמונה)
- ▶ SPRITE יכול להיות מוצג על הCANVAS. SPRITE יכול לשתף פעולה עם SPRITE ים קיימים אחרים על גבי אותו CANVAS



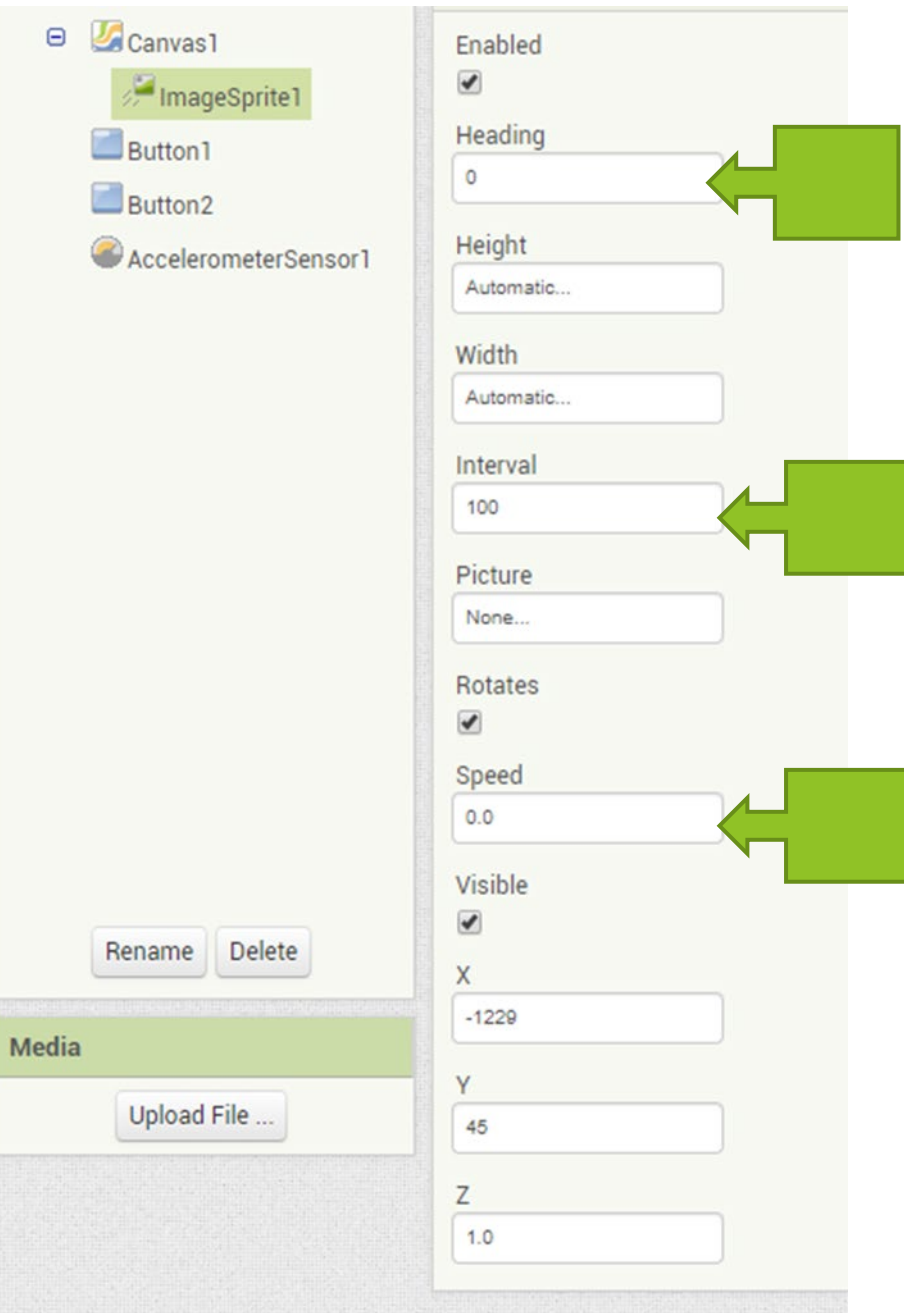
SPRITE - CANVAS



בחירת
תמונה

- ▶ ניתן להגדיר תמונה -
- ▶ התמונה צריכה להיות קטנה על מנת שתוכל לנוע על פני המסך
- ▶ כמובן, יש לטעון את התמונה בדרך שלמדנו לאפליקציה

SPRITE - CANVAS



▶ על מנת שהצורה שלנו תתחיל לזוז עלינו לקבוע לה 3 פרמטרים

▶ כיוון

▶ מהירות

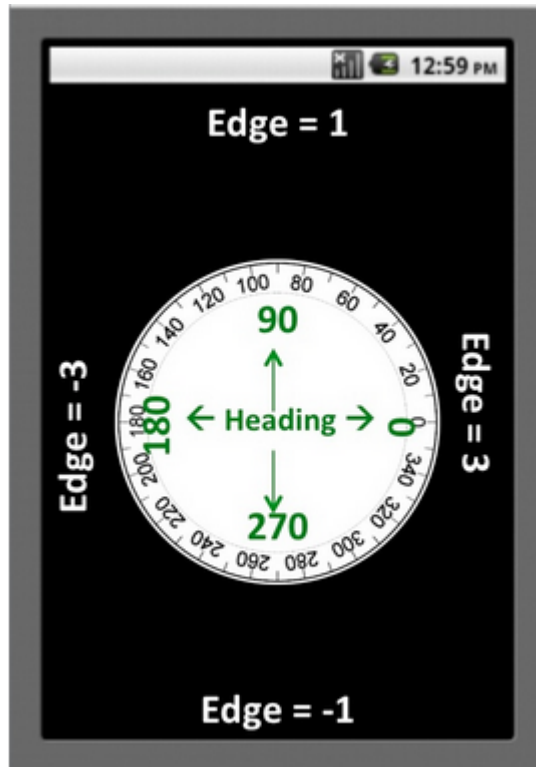
▶ INTERVAL - מרווח

▶ מיקום התחלתי

▶ בנוסף יש להדליק את ENABLED

▶ לכל צורה ניתן לקבוע כוון בו היא יכולה להתחיל לזוז

SPRITE - CANVAS



- ▶ פרמטר כוון מסמל כוון בו נעה התמונה
- ▶ כוון נמדד במעלות
- ▶ 0 מעלות הן בצידו הימני של המכשיר
- ▶ 90 מעלות למעלה ו180 בצד ימין

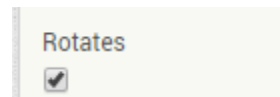
SPRITE - CANVAS

- ▶ פרמטר שני הוא פרמטר מהירות SPEED ופרמטר INTERVAL
- ▶ המהירות התנועה של גוף נמדדת ככמות כמה פעמים מצטיירת התנועה במרווח מסוים
- ▶ לכן יש להגדיר בו זמנית את SPEED ואת INTERVAL
- ▶ למשל אם INTERVAL שווה ל-50 ו-SPEED שווה ל-10, זה הוא שגוף שלנו יתקדם 10 פיקסלים כל 50 שניות

SPRITE - CANVAS

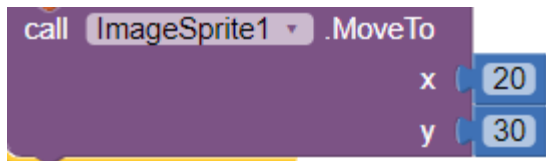
ROTATES קובע האם הצורה זזה לפי הכוון שהוגדר לה בHEADINGS ►

ROTATES מקבל רק שני ערכים TRUE או FALSE ►



SPRITE - MOVETO

- ▶ על מנת להתחיל את התנועה, נדרש למקם את SPRITE במיקום התחלתי כלשהו.
- ▶ לצורך כך יש לקרוא לMOVETO - תפקידו למקם את התמונה במקום מסוים



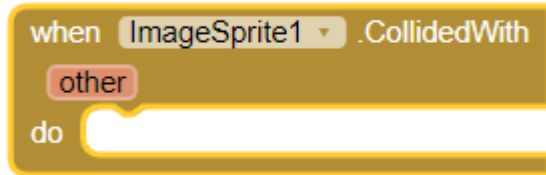
תרגיל מספר 3

- יש לפתח אפליקציה MOVEMOL שמזיזה תמונה של מול על פני המסך
- יש להעלות את התמונה מוקטנת של מול ולהתחיל להזיז אותו



SPRITE - אינטרקציה עם דמויות אחרות

- ▶ לכל SPRITE ניתן לבדוק האם הוא מתנגש או חוצה את הדרך של SPRITE אחר כלשהו
- ▶ לצורך כך קיימים לSPRITE אירועים נוספים
- ▶ אירוע APP INVERTOR - COLLEIDEDWITH יקרא לנו מייד כאשר SPRITE יתנגש עם גוף אחר - למשל עם SPRITE אחר



SPRITE - אינטרקציה עם דמויות אחרות

- ▶ ניתן להזיז את SPRITE בצורה אקראית על גבי המסך
- ▶ לצורך כך, בקריאה לMOVETO ניתן לבחור לו מיקום אקראי



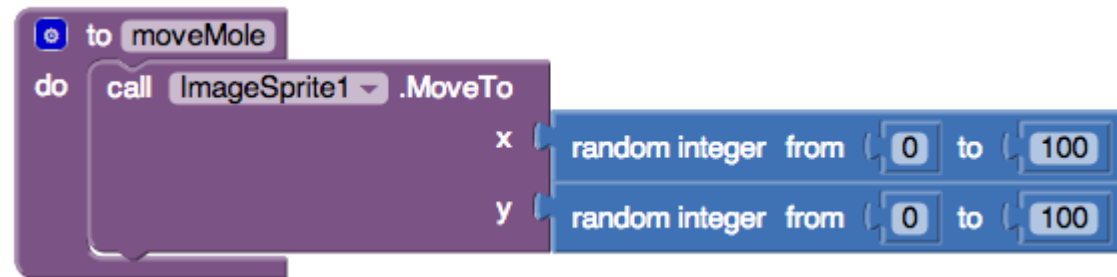
תרגיל מספר 4 - התנגשות בין ה-MOLE ים

- יש להוסיף לאפליקציה MOLE נוסף.
- יש למקם את שני ה-MOLEים בצורה אקראית על פני המסך
- במידה ו-1MOLE מתנגש עם 2MOLE יש לשנות את מיקום של MOLE 1 ל 0,0



פעולות

- ▶ APP INVERTOR מאפשר לפתח פעולה מסוימת ולקרוא לה בצורה סדרתית
- ▶ נניח שאנחנו מעוניינים להזיז MOLE
- ▶ קריאה להזזת MOLE חוזרת על עצמה כל פעם במקומות שונים באפליקציה
- ▶ אנחנו יכולים להגדיר פעול ולקרוא לה MOVEMOLE
- ▶ לאחר מכן נוכל פשוט לקרוא לפעולה MOVEMOLE בכל מיני מקומות בקוד שלנו



תרגיל מספר 5

יש לשנות את האפליקציה שפיתחתם בתרגיל מספר 4, להוסיף פעולה MOVEMOL ולקרוא לה במקומות שונים

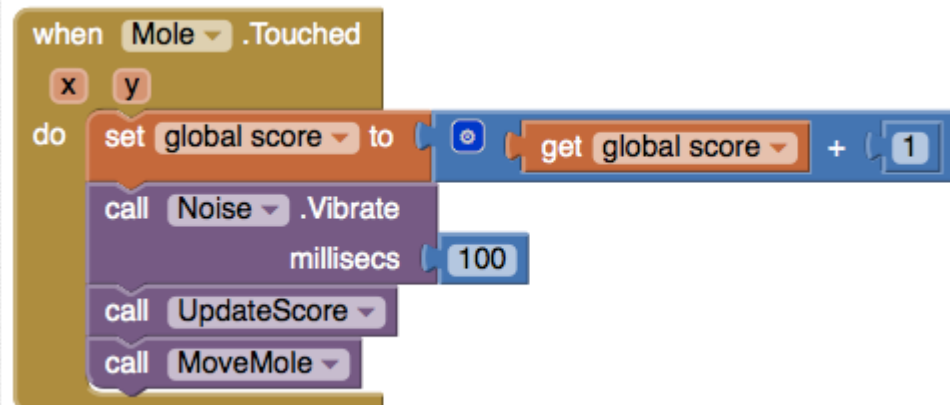


TOUCHED - אירוע SPRITE

▶ תפקידו של אירוע TOUCHED הוא לדווח מתי המשתמש בחר לעשות CLICK מעל הגוף SPRITE

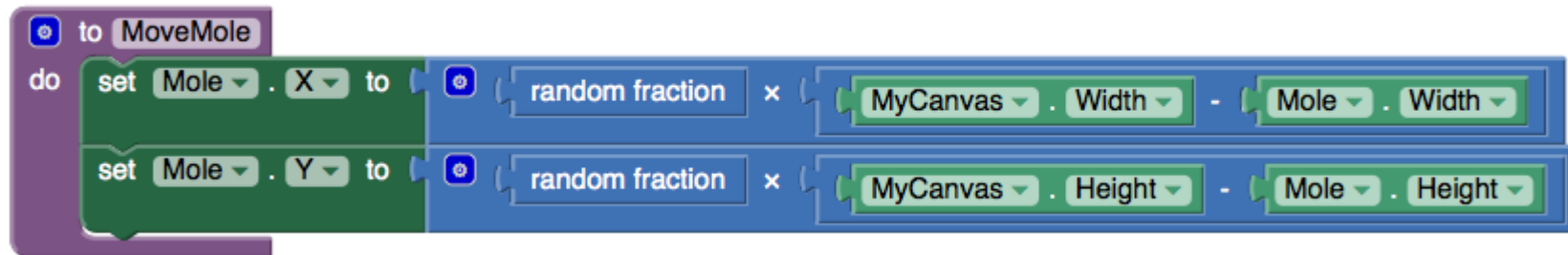
▶ במקרה הזה ניתן לעשות פעולה כלשהי עם ה- SPRITE - למשל להזיז אותו

▶ למשל, בדוגמא הזאת בלחיצה על MOLE, מתעדכנת התוצאה, המכשיר רוטט, והצורה זזה למקום רנדומלי אחר



פרויקט - תרגיל מספר 6

- יש לפתח אפליקציה MOVEMOL
 - האפליקציה מציגה את MOL שנע על פני המסך
 - למטה במסך מופיע TEXTBOX SCORE המציג את התוצאה הנוכחית
 - בכל פעם שהמשתמש לוחץ על MOL יש להקפיץ אותו למקום אחר ולהוסיף 1
- <https://appinventor.mit.edu/explore/ai2/molemask.html>



פיתוח אפליקציות

שיעור מספר 10

שיעור 10 - המשך גרפיקה ו-CANVAS

1. חזרה
2. פיתוח תנועת כדור
3. שימוש בטלפון - ביצוע שיחת טלפון
4. שימוש במצלמה

תרגיל מספר 1 - חזרה

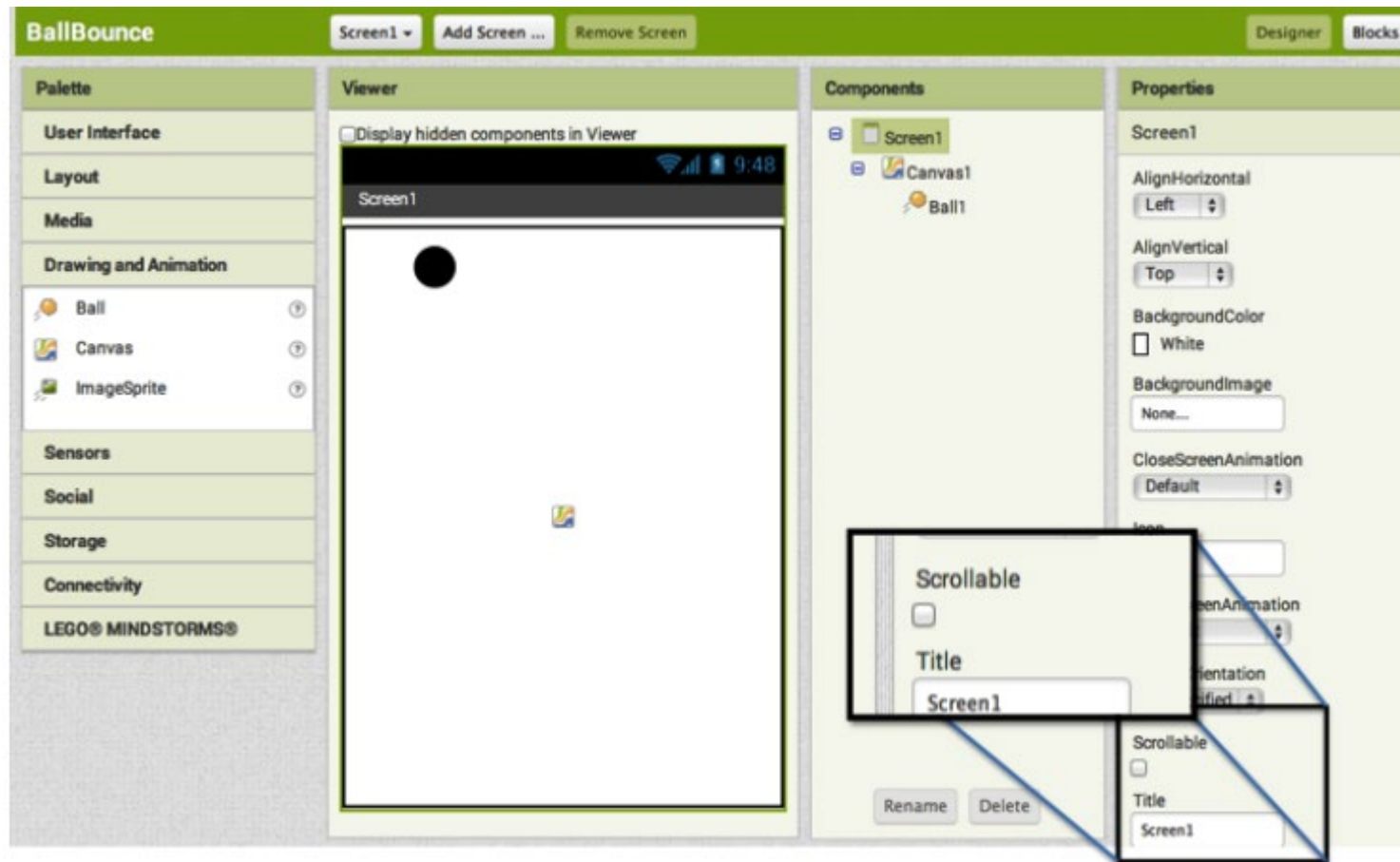
- יש לעבור על התרגילים שלמדנו בשיעור הקודם
- יש לבדוק שפרויקט MOVEMOL עובד

פיתוח תנועת כדור

- ▶ ב-CANVAS קיים אובייקט נוסף - כדור BALL
- ▶ תפקידו של הכדור כמובן הוא לנוע על פני ה-CANVAS
- ▶ BALL - הוא SPRITE - אובייקטים אחרים יכול להתנגש בו
- ▶ ניתן לבדוק האם הכדור נגע בקירות ה-CANVAS, ניתן לשנות את המהירות של הכדור, כוון שלו וכל שאר הפרמטרים

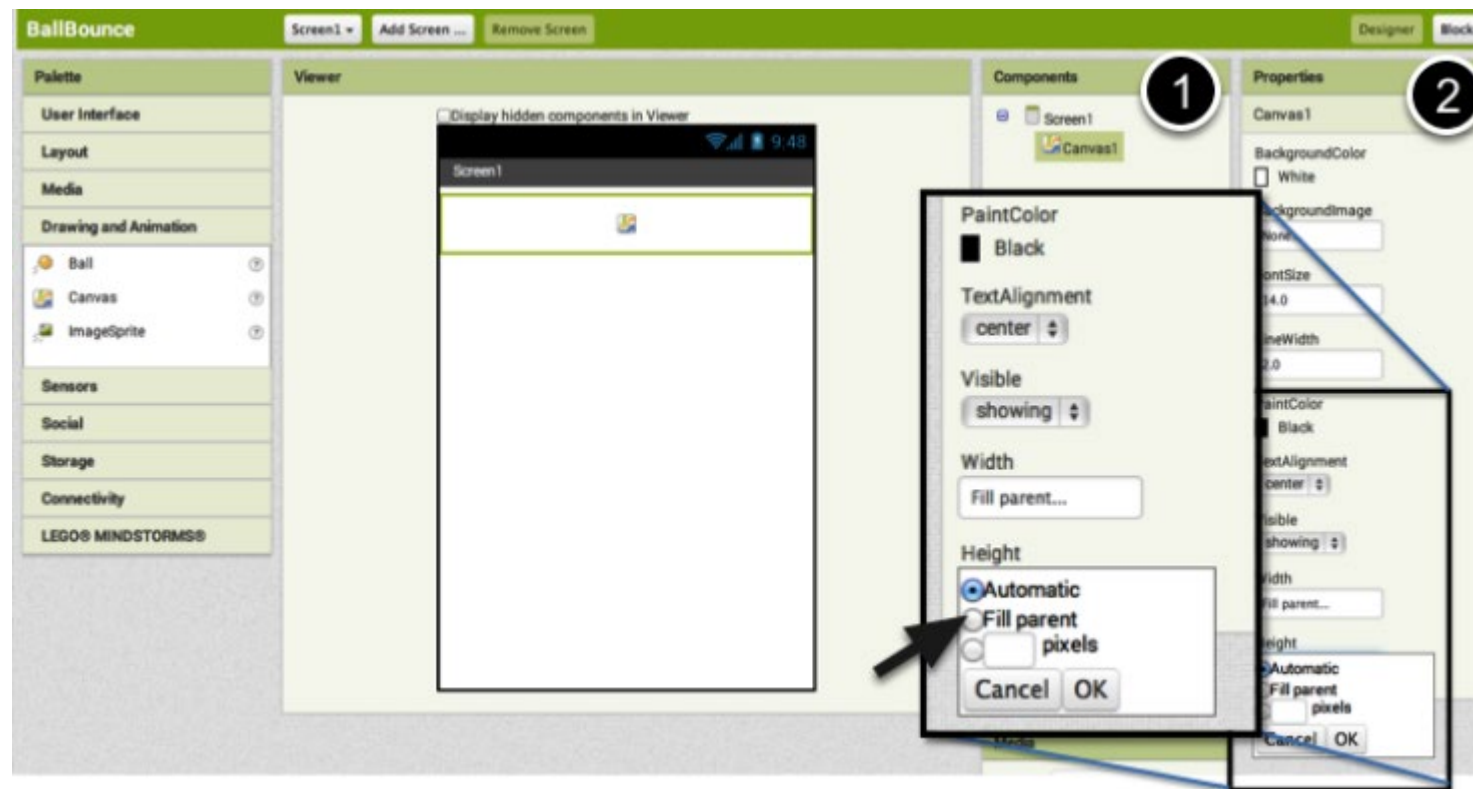
פיתוח תנועת כדור

- ▶ על מנת לפתח תנועת כדור שלא תתנגש בקיר, יש לחסום את SCROLLABLE של CANVAS
- ▶ למנוע מ CANVAS להתנגש בקיר



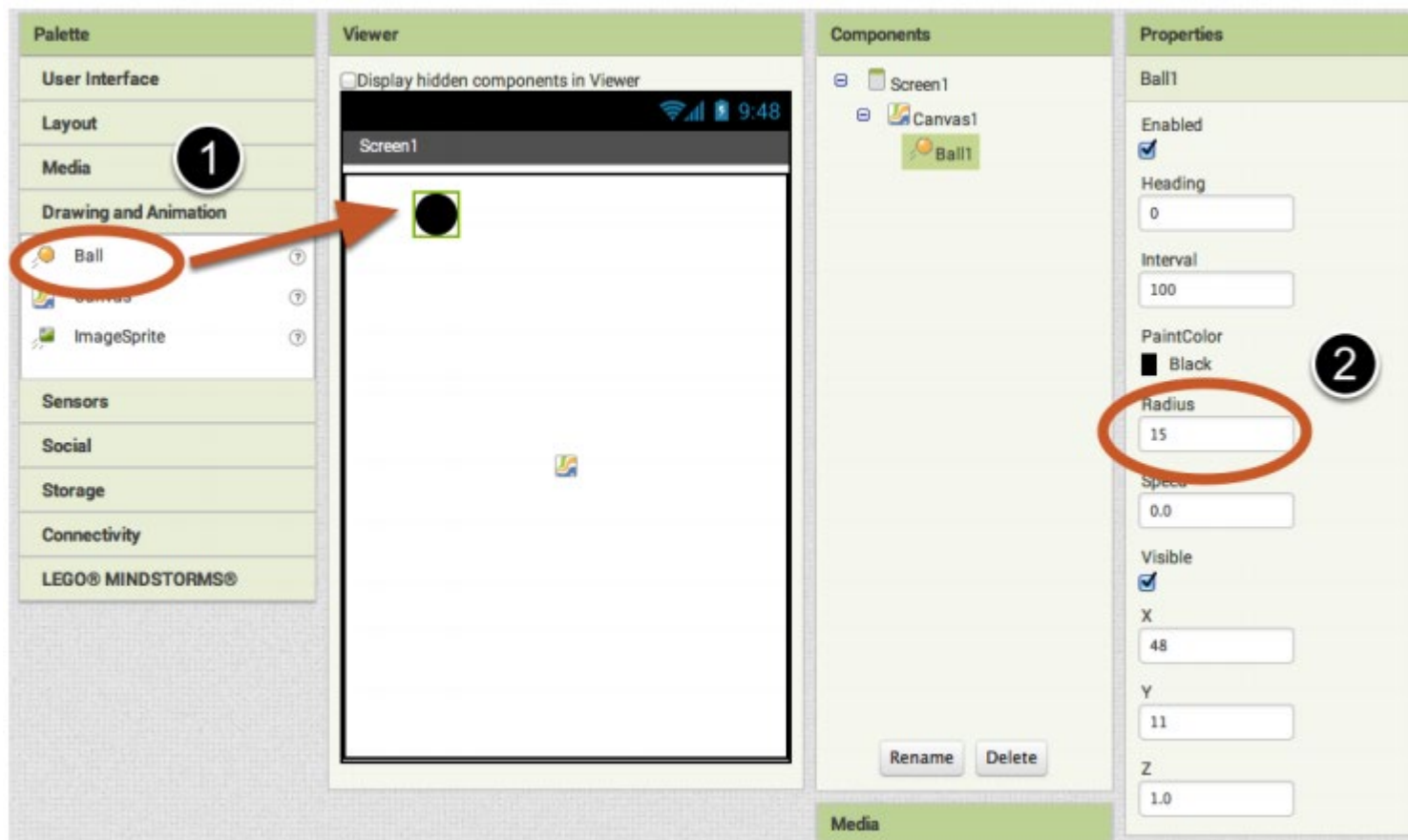
פיתוח תנועת כדור

בכל הוספת CANVAS יש לבצע FILL PARENT על מנת למלא את כל המסך



פיתוח תנועת כדור

יש לקבוע רדיוס 15 לכדור



פיתוח תנועת כדור

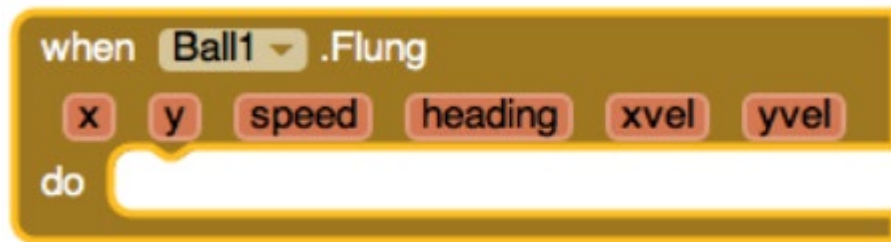
ל-BALL יש מספר פעולות

The image shows a Scratch-like development environment with a 'Blocks' panel on the left and a 'Viewer' panel on the right. In the 'Blocks' panel, the 'Ball1' object is selected and circled in red. The 'Viewer' panel displays several event-driven code blocks for 'Ball1':

- when Ball1 .CollidedWith other**: Annotated with 'בדיקת התנגשות עם אובייקט אחר' (Collision check with another object).
- when Ball1 .Dragged**: Includes input fields for startX, startY, prevX, prevY, currentX, and currentY. Annotated with 'גרירת כדור' (Ball dragging).
- when Ball1 .EdgeReached edge**: Annotated with 'בדיקת התנגשות עם קצה CANVAS' (Collision check with CANVAS edge).
- when Ball1 .Flung**: Includes input fields for x, y, speed, heading, xvel, and yvel.
- when Ball1 .No longer Colliding With**: Partially visible at the bottom.

פיתוח תנועת כדור FLUNG

- ▶ נבחר ב FLUNG - נגיעה של המשתמש בכדור (או בכלל במסך)
- ▶ כאשר משתמש מבצע FLUNG , מכשיר יכול לגלות לא רק מיקום הנגיעה, אלה גם כיוון והמהירות
- ▶ FLUNG מאפשר למשתמש להזיז את הכדור לאחד הצדדים



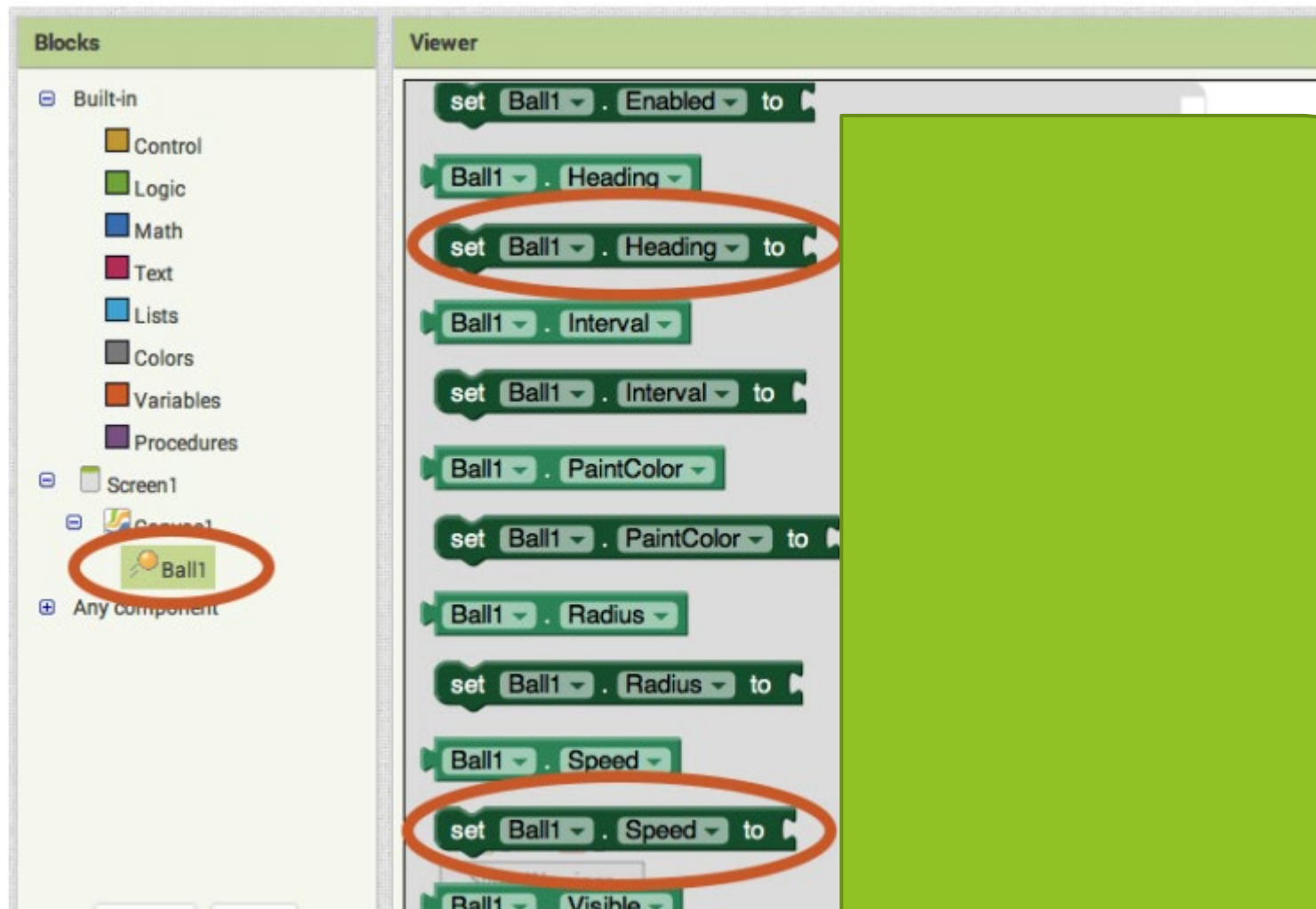
פיתוח תנועת כדור

- ▶ נבחר ב FLUNG - נגיעה של המשתמש בכדור (או בכלל במסך)
- ▶ כאשר משתמש מבצע FLUNG , מכשיר יכול לגלות לא רק מיקום הנגיעה, אלה גם כיוון והמהירות
- ▶ FLUNG מאפשר למשתמש להזיז את הכדור לאחד הצדדים



פיתוח תנועת כדור

- יש לבחור את הכדור ולעבור לBLOCKS
- ניתן לקבוע את כוון ואת מהירות הכדור



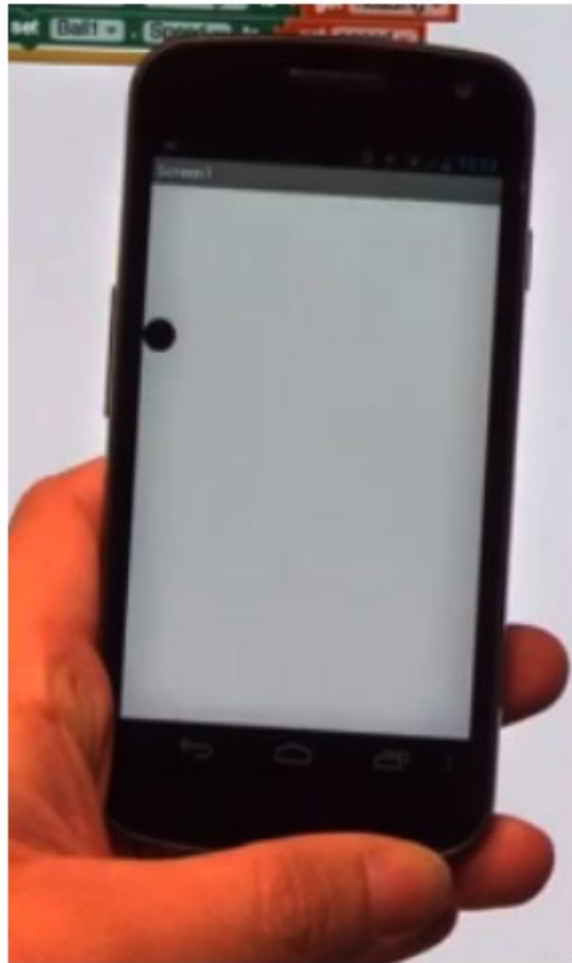
פיתוח תנועת כדור

FLUNG מאפשר לשנות את המהירות ואת הכוון הכדור לפי נגיעה של המשתמש



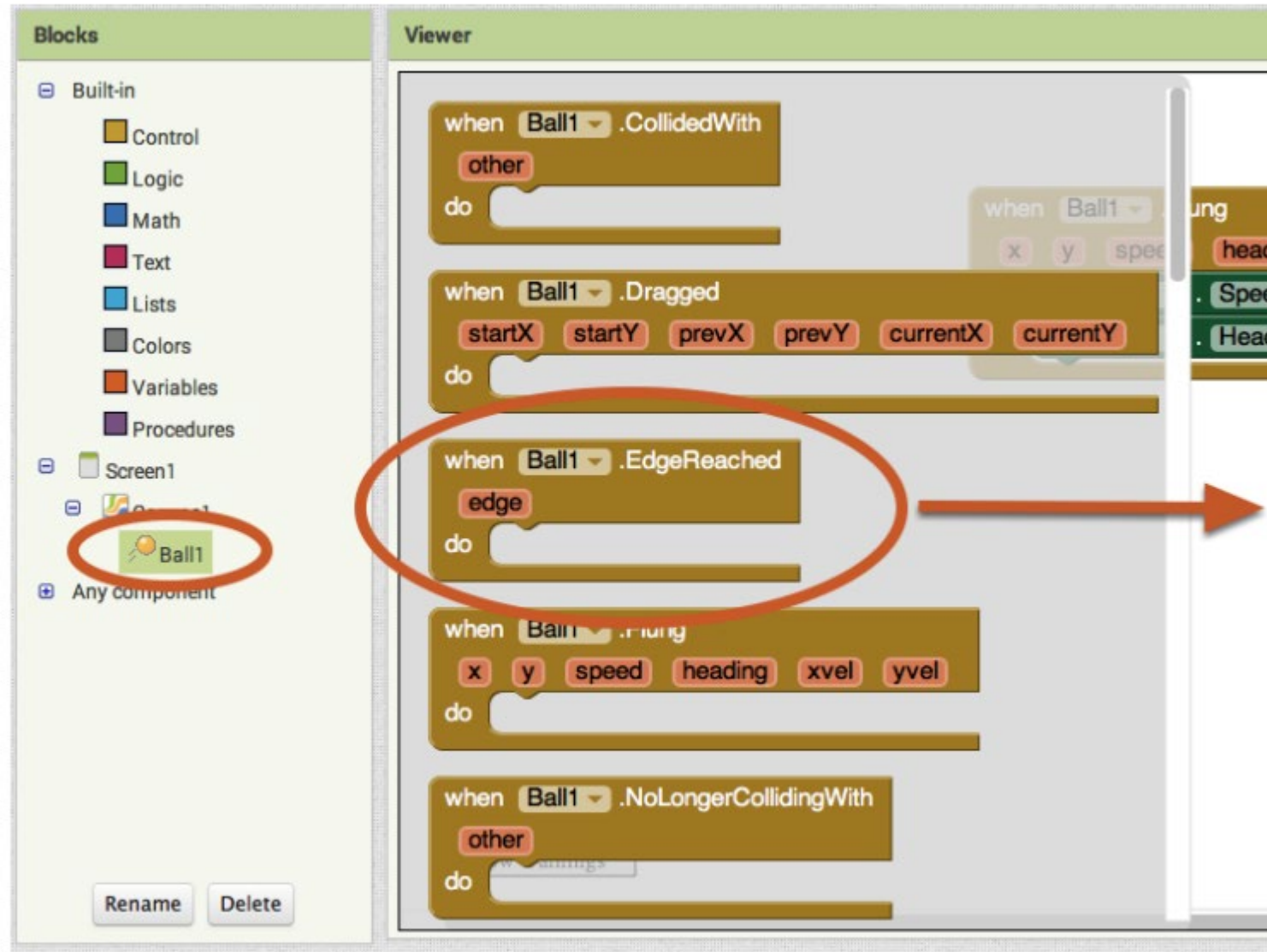
פיתוח תנועת כדור

ניתן להריץ את האפליקציה - תגלו שהכדור נמצא על צד אחד של המסך...



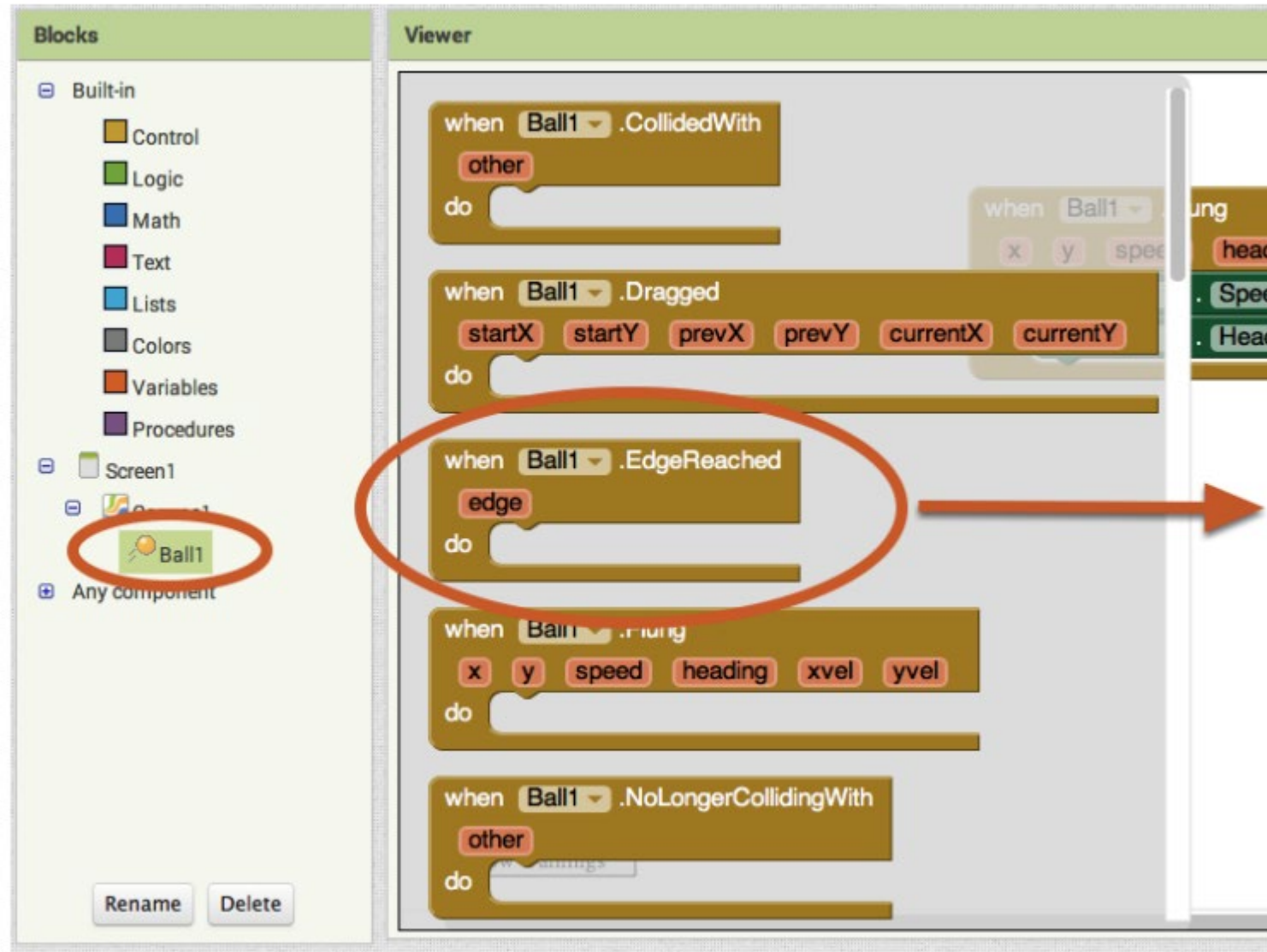
פיתוח תנועת כדור

- ▶ ניתן להריץ את האפליקציה - תגלו שהכדור נמצע על כוון אחד של המסך...
- ▶ על מנת להימנע מהבעיה, יש להשתמש באירוע EDGEREACHED - הגענו לקצה



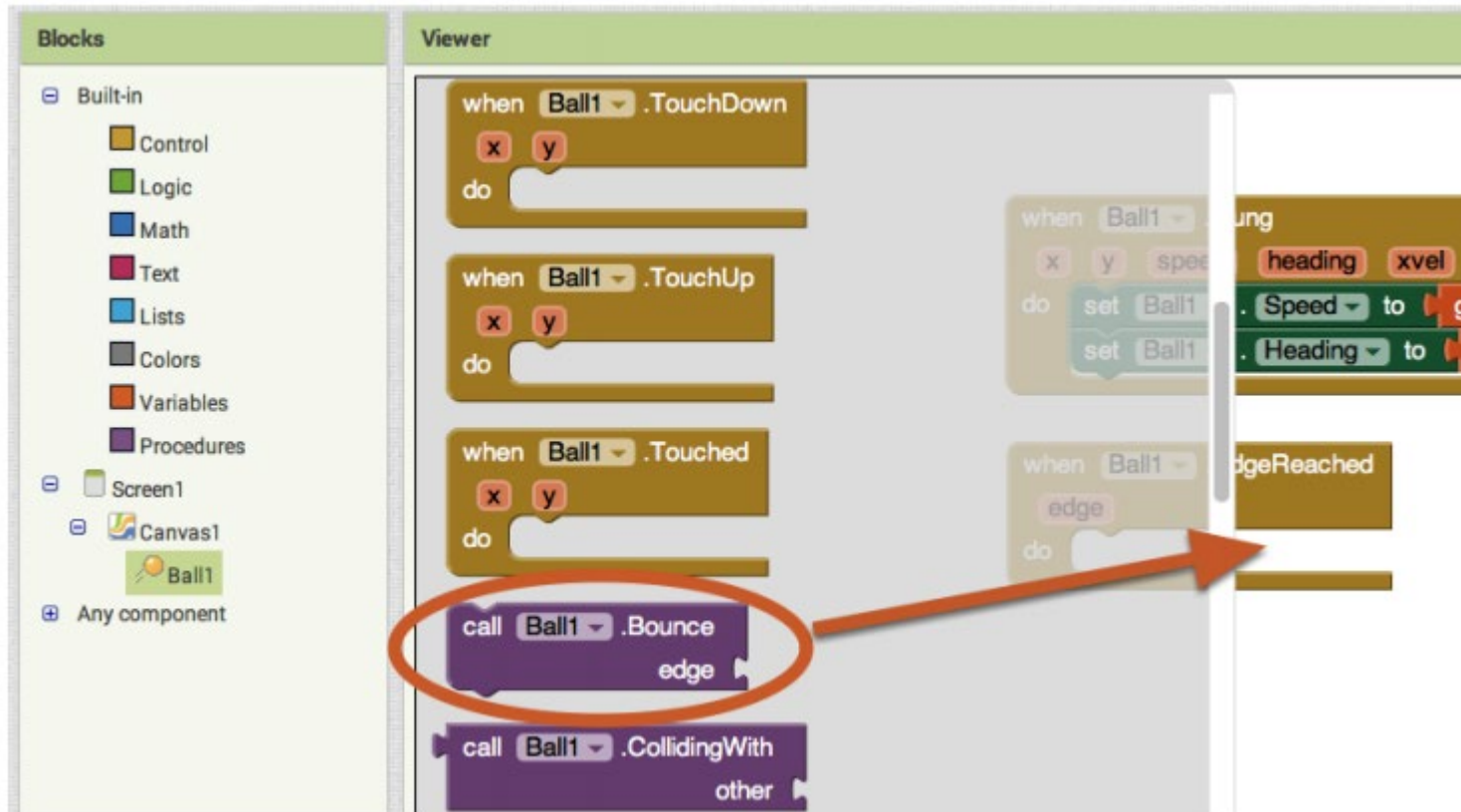
פיתוח תנועת כדור

- ▶ ניתן להריץ את האפליקציה - תגלו שהכדור נמצע על כוון אחד של המסך...
- ▶ על מנת להימנע מהבעיה, יש להשתמש באירוע EDGEREACHED - הגענו לקצה



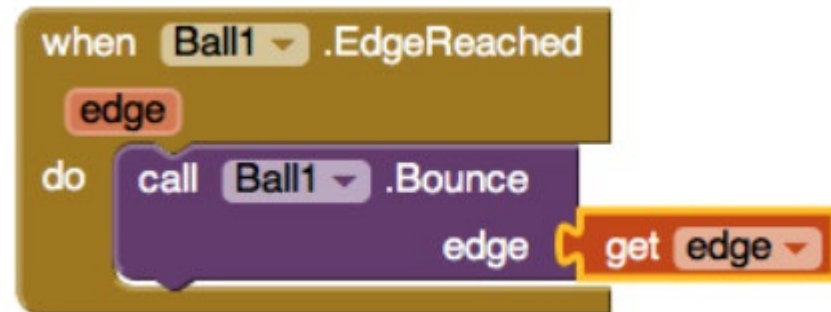
פיתוח תנועת כדור

BALL BOUNCE מקבלת כפרמטר EDGE הקצה ובעצם מבצע סיבוב של הכדור בהתאם למיקומו



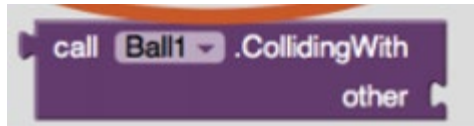
פיתוח תנועת כדור

- ▶ BALL BOUNCE מקבלת כפרמטר EDGE הקצה ובעצם מבצע סיבוב של הכדור בהתאם למיקומו
- ▶ עכשיו יש להפעיל אפליקציה ולעקוב אחרי הכדור



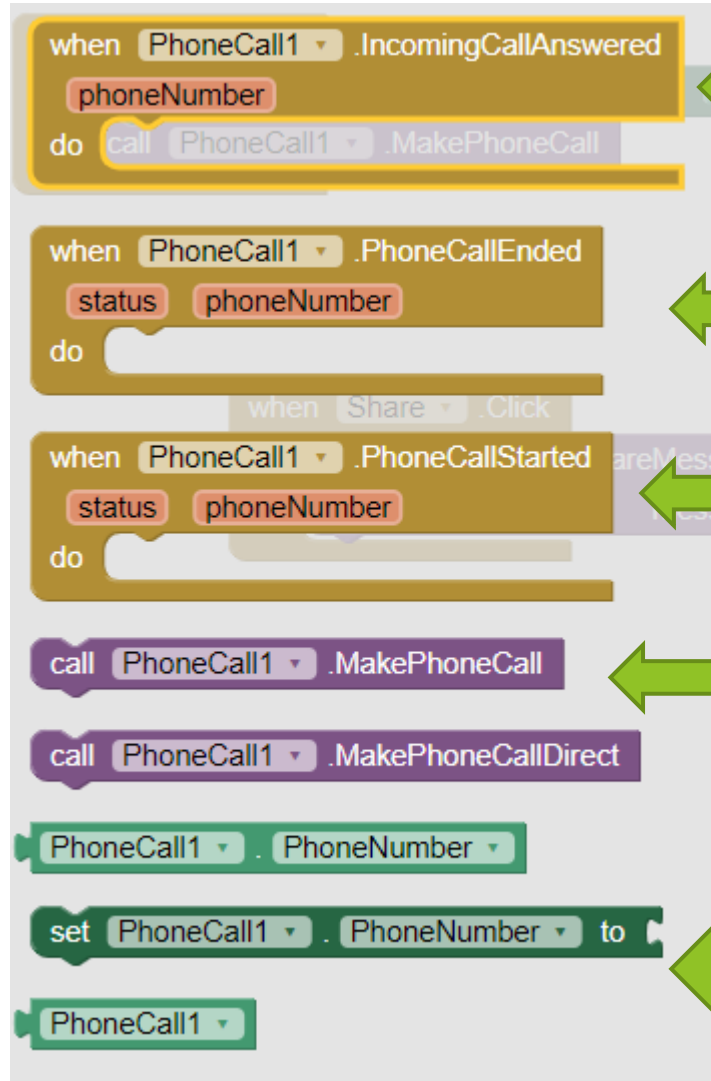
תרגיל 2 - פיתוח שלב 1 של הזזת כדור והתנגשות עם הקירות

- יש לפתח את שלב 1 לפי ההוראות ולוודא שהכדור זז לפי הצורך
- יש להוסיף תמיכה בהתנגשויות בין כדורים - במידה ויש התנגשות, יש לשנות כוון הכדורים
- יש להשתמש ב-COLLIDINGWITH - התנגשות עם...



ביצוע שיחת טלפון וכפתור SHARING

לרכיב PHONECALL יש מספר אירועים חשובים ופעולות חשובות



נקרא כאשר מגיע
שיחה נכנסת

נקרא כאשר שיחה
מסתיימת

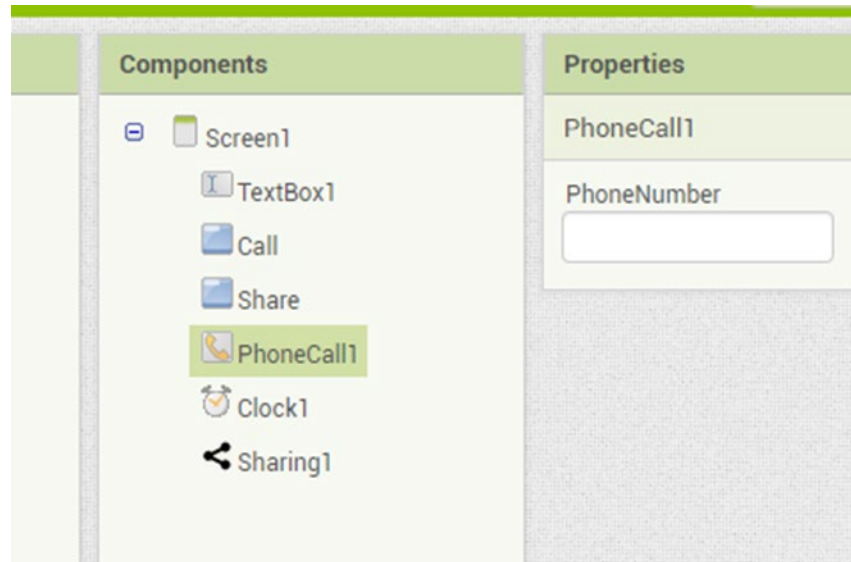
נקרא כאשר שיחה
מתחילה

מחייג על פי מספר
טלפון

משפיע - מחזיר או
משתנה את מספר טלפון

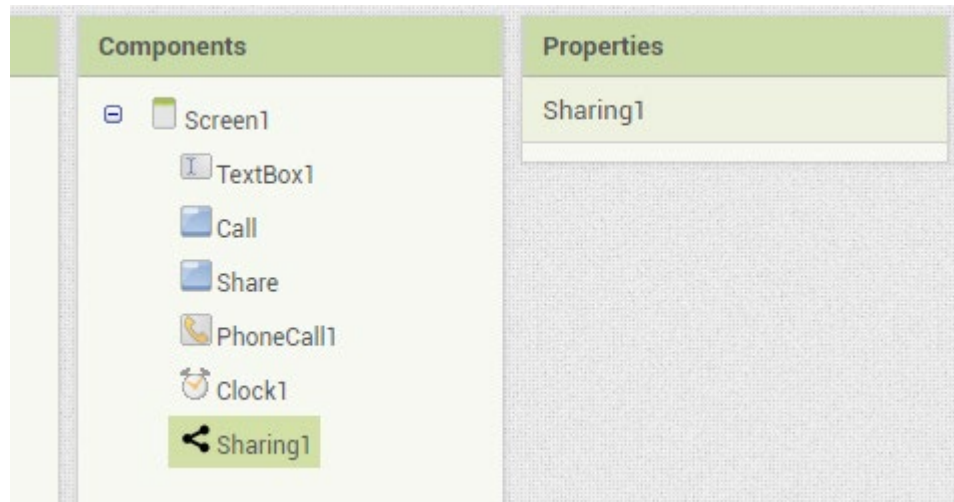
ביצוע שיחת טלפון וכפתור SHARING

בבחירת PHONECALL יש רק מאפיין אחד - מספר טלפון ►



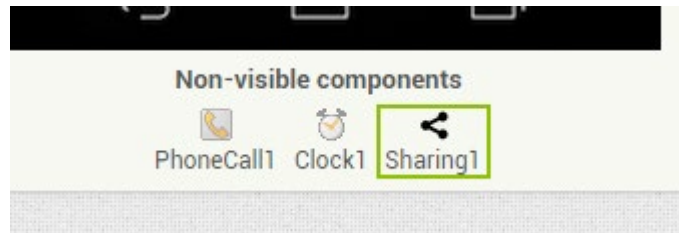
ביצוע שיחת טלפון וכפתור SHARING

- ▶ כפתור SHARING מאוד פופלארי בהרבה אפליקציות - מטרתו לשתף אל התמונה או טקסט שהוקלט באפליקציה , באפליקציות אחרות.
- ▶ המון אפליקציות עושות שימוש בכפתור זה.
- ▶ בבחירת SHARING לא קיימות אופציות מכוון שכל מכשיר מאפשר שיתוף בהתאם לאפליקציות שמותקנות בו



ביצוע שיחת טלפון וכפתור SHARING

- ▶ הן PHONECALL והן SHARING הם רכיבים בלתי נראים - INVISIBLE
- ▶ הרכיבים הללו מופיעים למטה בדף אבל הם בלתי נראים על ידי המשתמש



תרגיל מספר 3

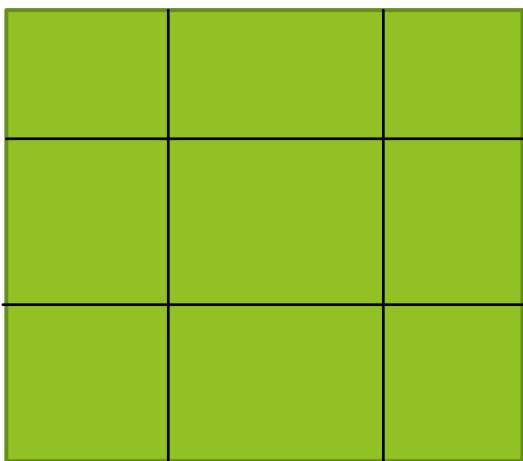
- ▶ כתוב אפליקציה, המבצעת שיחת טלפון
- ▶ משתמש יכול להזין מספר טלפון וללחוץ על הכפתור OK לביצוע השיחה
- ▶ בנוסף כאשר מקבל משתמש שיחה נכנסת, האפליקציה מדפיסה את מספר הטלפון הנכנס
- ▶ בנוסף אפליקציה תבצע SHARE לטלפון שהתקבל (למשל לתוך ה WHATSAPP)

תרגיל מספר 4 תוספת לפרויקט פיצה

- ▶ יש להוסיף לאפליקציה הזמנת פיצה את האפשרות לדבר עם בעל ההזמנה, על מנת לוודא את פרטי ההזמנה
- ▶ בנוסף יש להוסיף אפשרות לשטף את תמונה של פיצריה בוטסאפ ופייסבוק

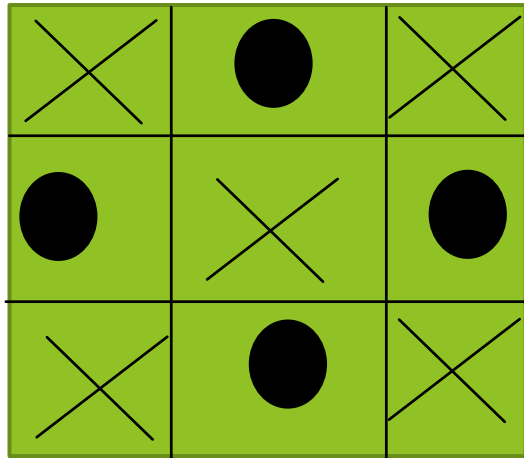
תרגיל מספר 6 יש לפתח אפליקציה למשחק איכס עיגול

יש קודם כל לצייר לוח עם 9 ריבועים



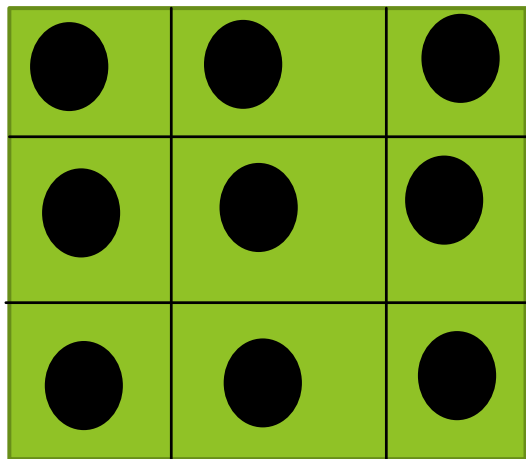
יש לפתח אפליקציה למשחק איכס עיגול

יש לצייר X ו) בכל מקום בלוח - יש לבדוק אפשרות לצייר X בכל מקום ובלוח ולאחר מכן לבדוק אפשרות לצייר O בכל מקום בלוח



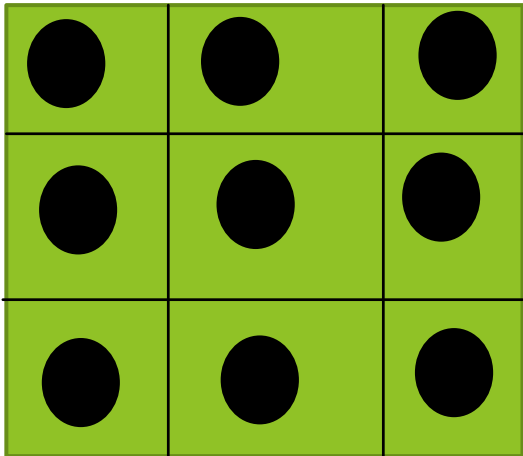
פרויקט אופציונלי - מורכב - טיק טק טו

- יש להוסיף אפשרות לצייר את X במקום בו משתמש לחץ על קליק
- אותו דבר לעשות עם עיגול



פתיחת פרויקט קיים - איכס עיגול

- יש להוסיף אפשרות לצייר את X במקום בו משתמש לחץ על קליק
- אותו דבר לעשות עם עיגול



איקס עיגול - לוגיקה

- ▶ בשלב ב - נצייר איקסים או עיגולים על פני הלוח.
- ▶ יש לגלות בדיוק מיקום שבו ביצע המשתמש קליק ולמקם את ה X או O במקום הנכון
- ▶ בנוסף יש להחליף טור ובכל קליק יש לצייר או X או O

X	X	X
X	X	X
X	X	X

פיתוח אפליקציות

שיעור מספר 11

שיעור 11 - פיתוח פרויקט ולימוד דברים נוספים

1. WEBVIEW
2. שימוש במצלמה
3. הפצת APK של אפליקציה
4. TINYDB - שמירת בבסיס הנתונים

תרגיל מספר 1 - תרגיל חזרה

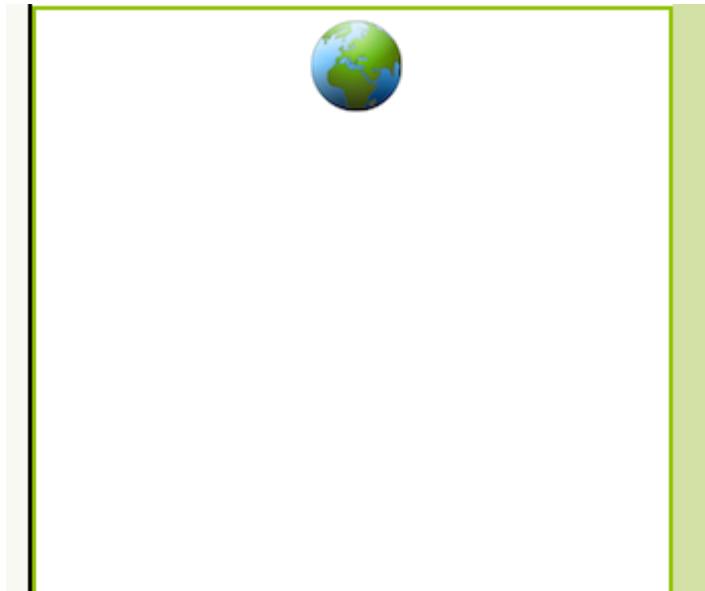
- ▶ יש להוסיף לתרגיל הזזת כדור, שפיתחנו בשיעור הקודם, אפשרות להתנגש עם הצורות האחרות
- ▶ יש להוסיף עוד 5 צורות אחרות על הגבי המסך
- ▶ יש לאפשר התנגשות בין צורות - במידה והצורות מתנגשות יש לשנות את כוון התנועות

WEBVIEW

- ▶ האפליקציה שלנו יכולה להציג גם דפדפן או אתר כלשהו שפיתחנו
- ▶ המשתמש לא תמיד יכול ומסוגל להבחין בכך.
- ▶ למשל, האם הבחנתם שYNET או WALLA הם בעצם אתרים מוסתרים בדפדפנים.
- ▶ דרך פיתוח זאת מקלה מאוד על החברות - הם לא יצטרכו לפתח בו זמנים אפליקציה וWEB

WEBVIEW

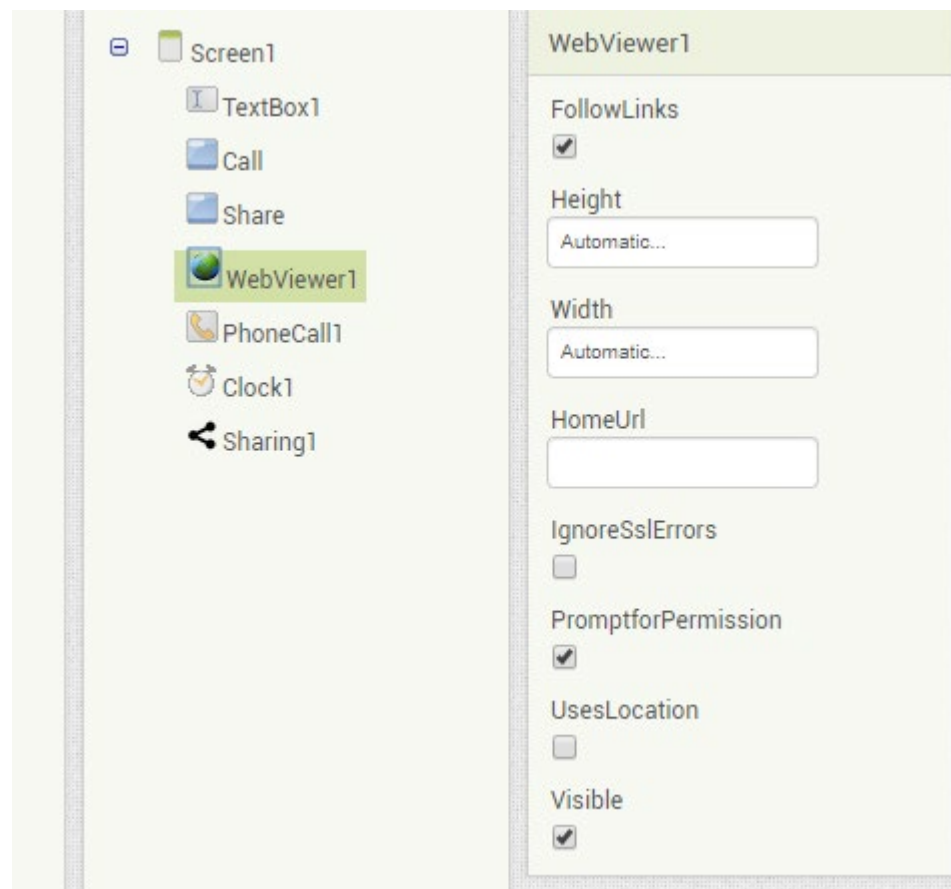
- ▶ האפליקציה שלנו יכולה להציג גם דפדפן או אתר כלשהו שפיתחנו
- ▶ המשתמש לא תמיד יכול ומסוגל להבחין בכך.
- ▶ למשל, האם הבחנתם שYNET או WALLA הם בעצם אתרים מוסתרים בדפדפנים.
- ▶ דרך פיתוח זאת מקלה מאוד על החברות - הם לא יצטרכו לפתח בו זמנים אפליקציה וWEB



WEBVIEW

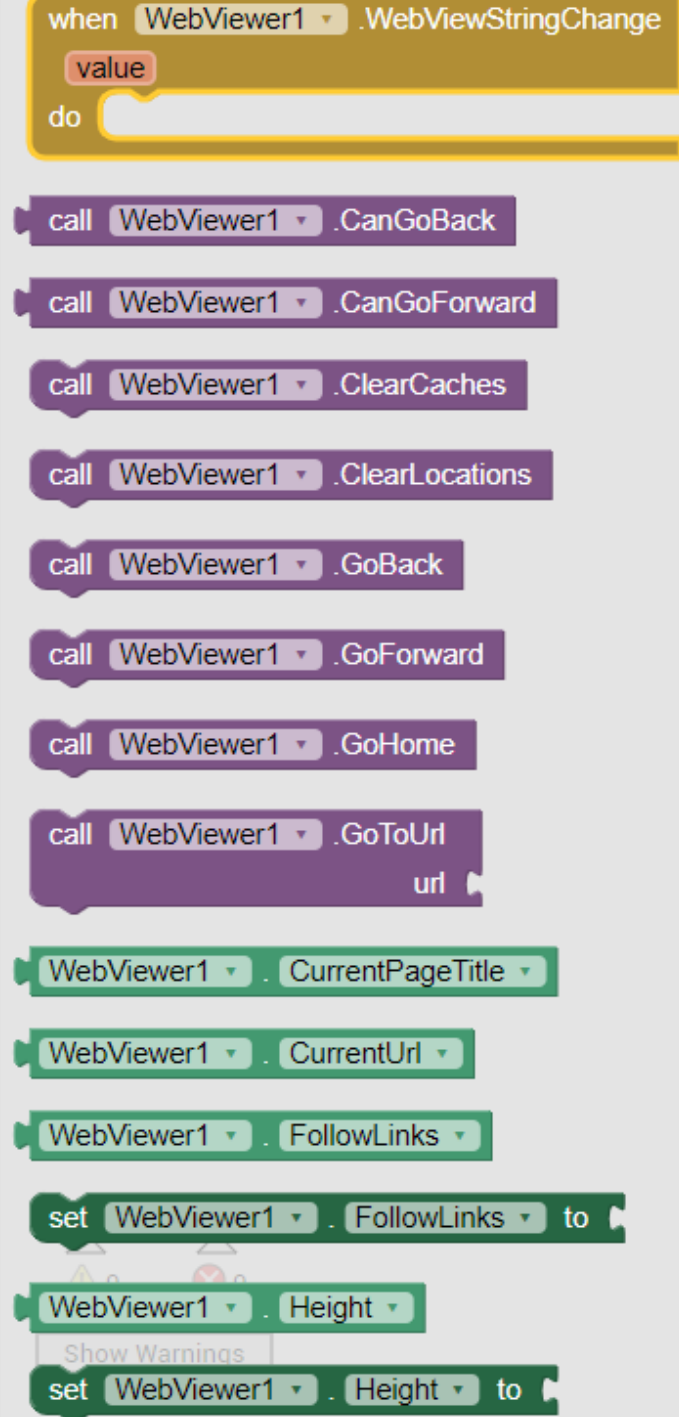
ניתן להזין HOMEURL - המסמל לינק לאתר בו ניתן לצפות

כמובן ניתן לקבוע וגודל ורוחב של הרכיב



WEBVIEW

- מכיל מספר רב פעולות הנוגעות לעבודה של דפדפן
- כגול CLEAR CACHE ,GO FORWARD , GO BACK ועוד
- GOTOURL - מכילה פקודה לדפדפן באפליקציה לפתוח אתר
- CURRENT PAGE TITLE - מעדכן את הכותרת של דף הנוכחי



תרגיל מספר 2

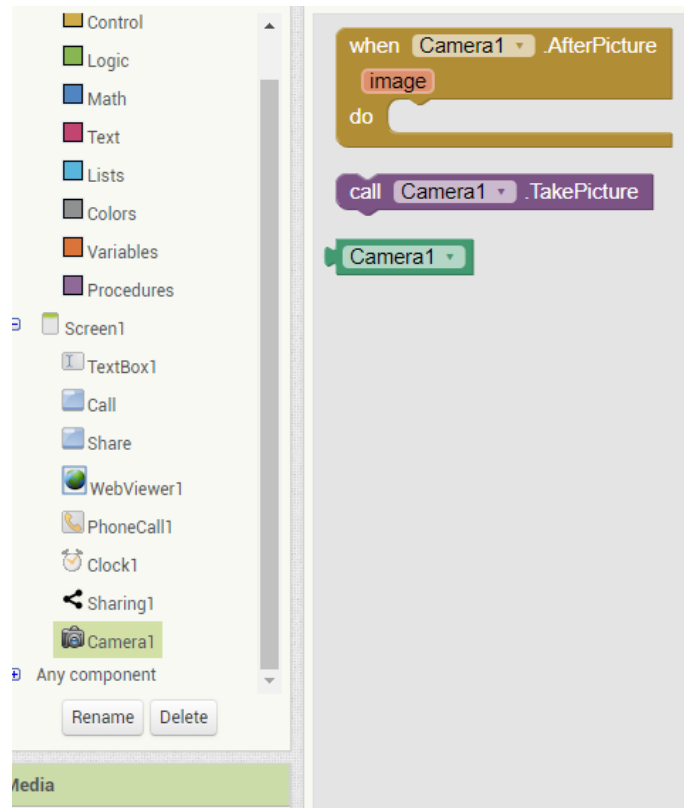
- ▶ פתח אפליקציה המציגה אתר YNET
- ▶ ובלחיצה על כפתור עוברת ומציגה אתר WALLA
- ▶ ובלחיצה על עוד כפתור מציגה אתר CNN
- ▶ שימו לב - על מנת להציג אתר יש לכתוב כתובת האתר בפורמט <http://www.ynet.co.il>

מצלמה ושימוש במצלמה

- ▶ על מנת לצלם באמצעות המצלמה יש להפעיל אותה ולבצע צילום.
- ▶ במידה ואתם עובדים עם הסימולטור - צילום אלול שלא לעבוד ואלול גם להפעיל את המצלמה של נייד שלכם

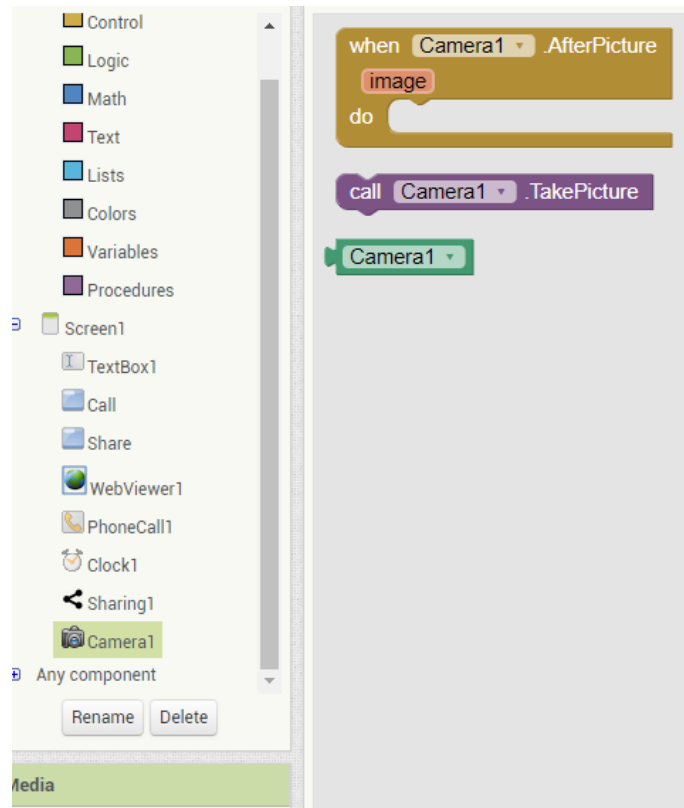
מצלמה ושימוש במצלמה

- ▶ המצלמה מכילה רק את האירוע בודד AFTER PICTURE
- ▶ והמצלמה יש לה פעולה בסיסית TAKE PICTURE
- ▶ את התמונה מקבלים מAFTER PICTURE



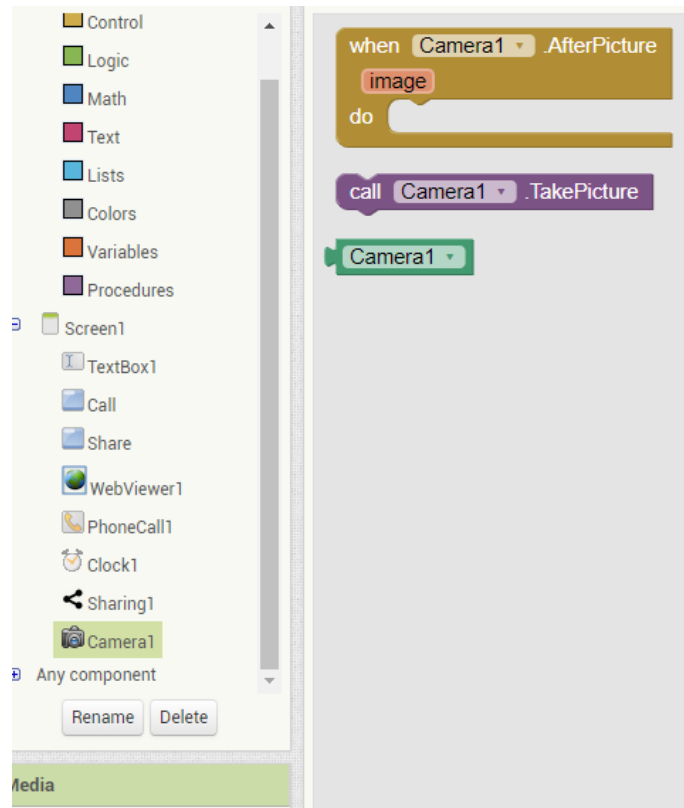
תרגיל מספר 3 - מלצמה ושימוש בה

- ▶ כתוב אפליקציה העוזרת לך לצלם בוק עצמי
- ▶ צלם את התמונה של עצמך



תרגיל מספר 3 - מלצמה ושימוש בה

- ▶ כתוב אפליקציה העוזרת לך לצלם בוק עצמי
- ▶ צלם את התמונה של עצמך
- ▶ הצג את התמונה בתצוגת IMAGE



הכנה והפצת פרויקטים ב־APP INVERTOR

▶ במהלך הקורס פיתחנו מספר פרויקטים גדולים

▶ ניהול חנות פיצה

▶ אפליקצייה מצגת תמונות

▶ משחקים

▶ העברת תנועת כדור

▶ טיק טק טו

יש לפתוח אפליקציה שפיתחתם

MIT APP INVENTOR

Projects ▾ Connect ▾ Build ▾ Help ▾

CallPhone Screen1 ▾ Add Screen ...

App (provide QR code for .apk)
App (save .apk to my computer)

Palette

User Interface

- Button ?
- CheckBox ?
- DatePicker ?
- Image ?
- Label ?
- ListPicker ?
- ListView ?
- Notifier ?
- PasswordTextBox ?
- Slider ?
- Spinner ?
- Switch ?
- TextBox ?
- TimePicker ?
- WebView ?

Viewer

☐ Display hidden components in Viewer

Screen1

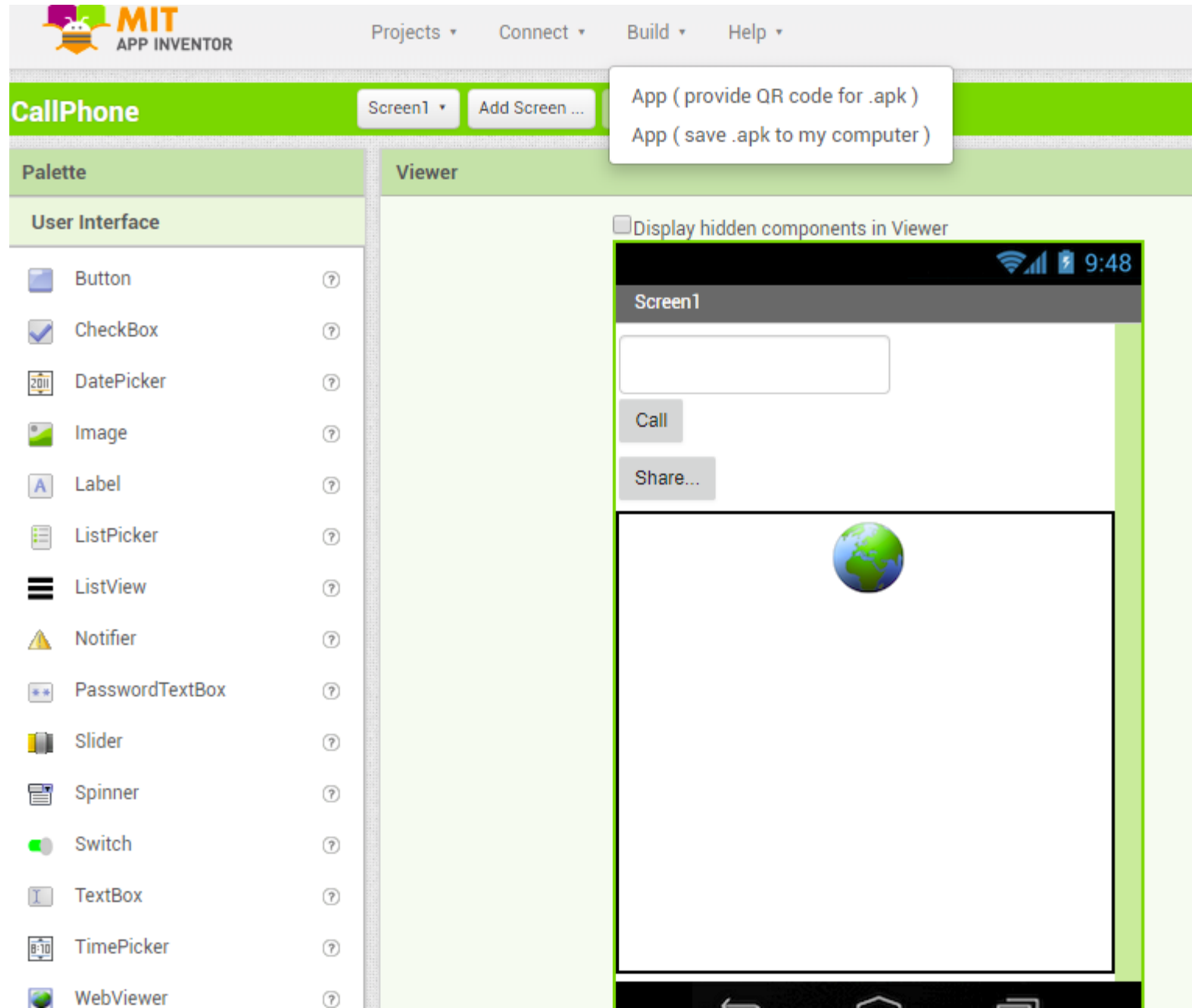
9:48

Call

Share...

WebView

יש לפתוח חוצץ BUILD ולבחור אופציה SAVE APK TO MY COMPUTER



יש לפתוח חוצץ BUILD ולבחור אופציה SAVE APK TO MY COMPUTER

- ▶ יוצר קובץ APK - קובץ APK זה קובץ אשר מכיל את כל ההתקנה של אפליקציה
- ▶ יש לשלוח את הקובץ למייל
- ▶ יש לפתוח את הקובץ דרך מכשיר הסלולארי
- ▶ במכשיר הסלולארי יש להתחבר למייל ולבצע התקנה של האפליקציה
- ▶ מהיום האפליקציות הללו ישמרו במכשיר הטלפון שלכם!!!

תרגיל מספר 4

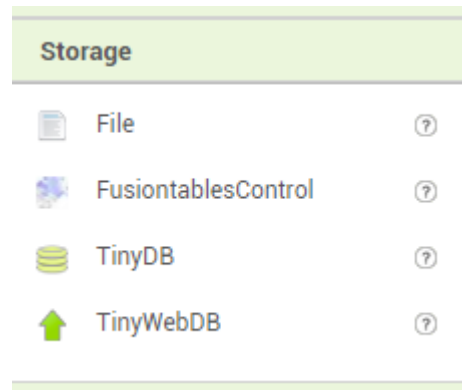
יש לשלוח את כל האפליקציות שפיתחתם ולהתקין אותם במכשיר ►

TINYDB שמירת נתונים לבסיס הנתונים

- ▶ עד כה בכל פעם בהם האפליקציה נסגרה, הנתונים לא נשמרו
- ▶ אבל APP INVERTOR כן מאפשר את שמירת הנתונים
- ▶ הדרך בה פועל APP INVERTOR היא דרך באמצעות DATABASE
- ▶ DATABASE זאת התוכנה שמאפשרת כתיבת נתונים של התוכנה לקובץ בפורמט מיולד
- ▶ שרות איתו אפשר להשתמש הוא שרות TINYDB

TINYDB שמירת נתונים לבסיס הנתונים

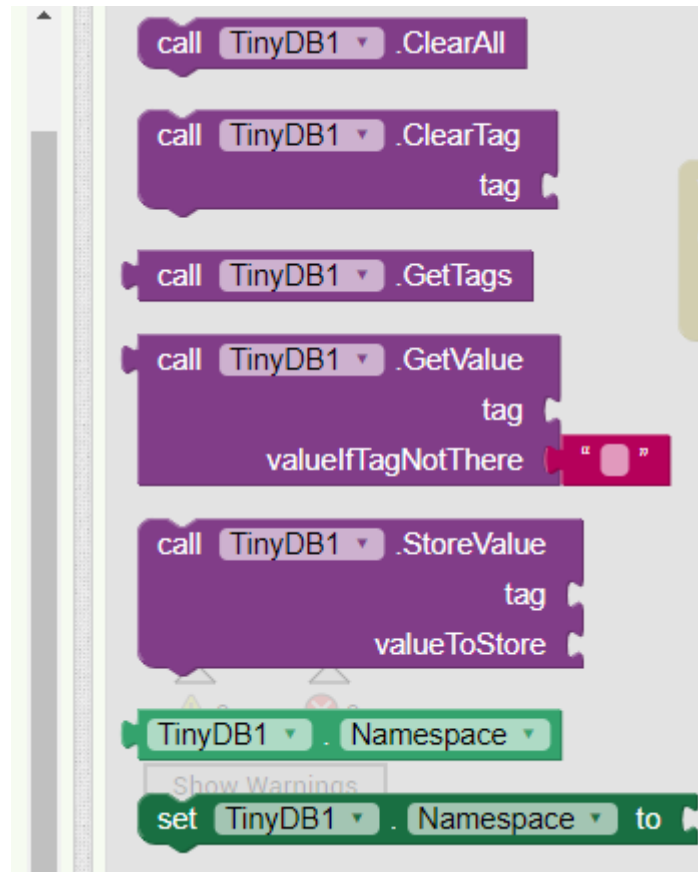
- יש לבחור את האופציה TINYDB
- יש לגרור את TINY DB על גבי המסך



TINYDB שמירת נתונים לבסיס הנתונים

- ▶ על מנת להבין וללמוד את הנושא, נפתח אפליקציה אשר מאפשר שמירת NOTES
- ▶ NOTES הם מילים קצרות המכילות כל מיני הודעות שרצינו לשמור לעצמינו
- ▶ מספר טלפון, חשבון בנק ועוד
- ▶ לא נראה כאשר האפליקציה תסגר שההודעות הללו ימחקו ולכן נשמור אותם ונאפשר למשתמש לטעון אותם

TINYDB שמירת נתונים לבסיס הנתונים



- פעולות על DB מאוד מאוד פשוטות
- CLEARALL - מוחק את כל הנתונים מהטבלה
- GETVALUE - טוען את הערך השמור בטבלה
- STOREVALUE - שומר את הערך בטבלה

יש לבנות את המסך הבא



TINYDB שמירת נתונים לבסיס הנתונים

נבצע אתחול לרשימת הודעות

initialize global notes to create empty list

when SubmitButton.Click

do if not is empty EntryTextBox.Text

then

add items to list list

item EntryTextBox.Text

set NotesLabel.Text to call listToText

list

set EntryTextBox.Text to

call TinyDB1.StoreValue

tag "notes"

valueToStore get global notes

Now we store the updated list persistently, in the database, so it will be there even if app is closed. The tag names the data. The valueToStore is the actual list.

to listToText list result...

when Screen1.Initialize

do set global notes to call TinyDB1.GetValue

tag "notes"

valueIfTagNotThere

create empty list

set NotesLabel.Text to call listToText

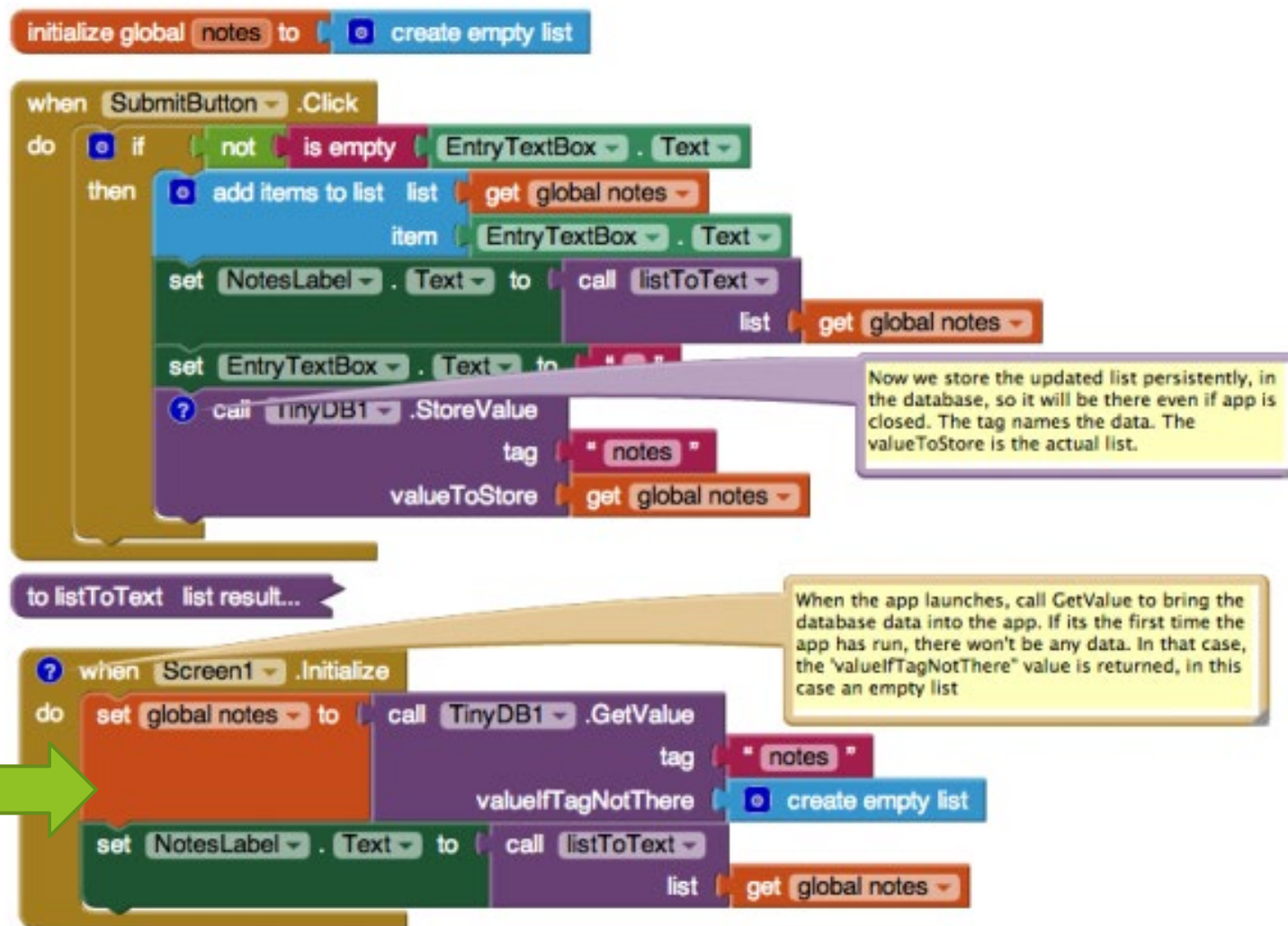
list

When the app launches, call GetValue to bring the database data into the app. If its the first time the app has run, there won't be any data. In that case, the 'valueIfTagNotThere' value is returned, in this case an empty list

בלחיצה על
הכפתור נעתיק
את ההודעת לDB

שמירת ההודעת
בבסיס הנתונים
DB

TINYDB טעינה לבסיס הנתונים



טעינת הנתונים
מתוך בסיס
הנתונים

פרויקט הזמנת פיצה

- יש לאחסן את מחיר הפיצה אשר הזמנתם בבסיס הנתונים
- יש לאפשר טעינת המחיר בפתיחה מחודשת של האפליקציה

סיום פרק ה-APPINVERTOR

- ▶ כאן אנו מסיימים את הפרק ב-APP INVERTOR בקורס
- ▶ בשיעור האחרון נלמד נושא אחר - APPSGEYSER

פיתוח אפליקציות

שיעור מספר 12

שיעור 12 - APPSGEYSER

1. מה זה APPSGEYSER?

2. דרך ההתחברות

3. בניית אפליקציה

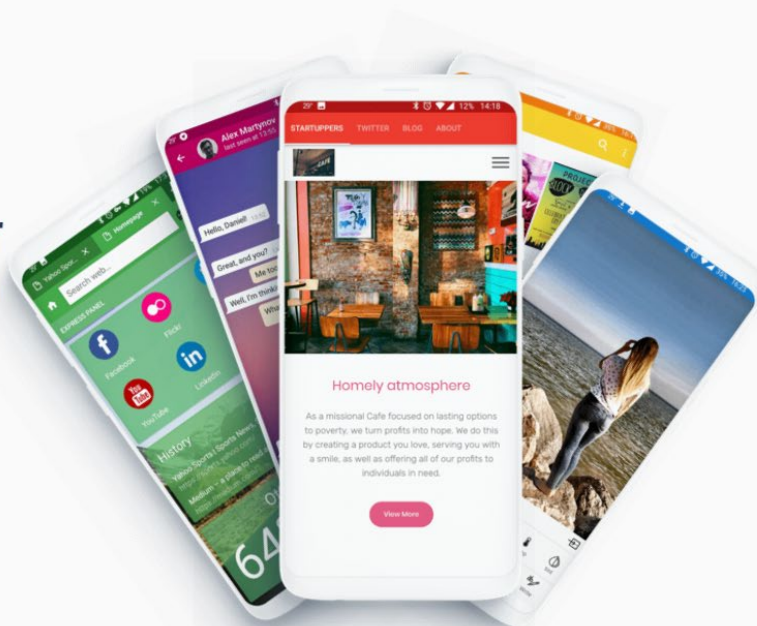
APPS GEYSER

- ▶ היום בשיעור נלמד תוכנה נוספת, איתה ניתן לפתח אפליקציות באנדרואיד
- ▶ תוכנת ה-APPS GEYSER
- ▶ התוכנה מאפשרת בנייה מהירה מאוד של אפליקציות

The Fastest and The
Simplest **App Builder**
for your app creation

— Without any coding and fees

CREATE NOW FOR FREE



APPS GEYSER

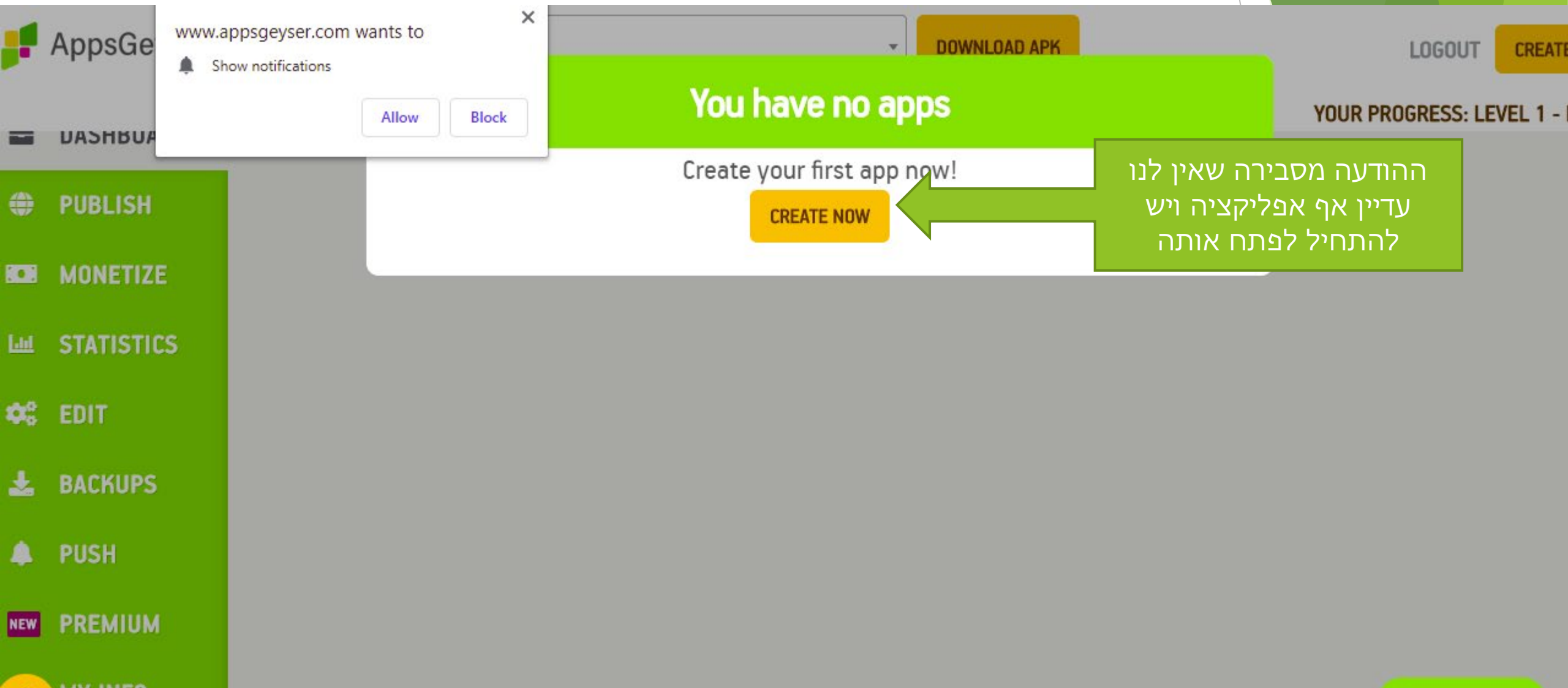
- ▶ בעבודה עם APP GEYSER אין צורך בשום רקע בתכנות - אפליקציות באתר כבר מוכנות
- ▶ באתר יש מאגר גדול מאוד של אפליקציות
- ▶ כל מה שמשתמש צריך לעשות זה
 - ▶ לבחור את אפליקציית בסיס
 - ▶ להכין מראש הרבה תמונות
 - ▶ להזין תיאור במקומות המיועדים לכך

התחברות לAPP GEYSER

- ▶ לצורך ההתחברות לAPP GEYSER יש ליצור חשבון באתר
- ▶ את החשבון ניתן לקשר לחשבון GOOGLE שלכם
- ▶ לצורך ההתחברות יש לפתוח GOOGLE CHROME הדפדפן
- ▶ יש לכתוב <https://www.appsgeyser.com>

התחברות ראשונה

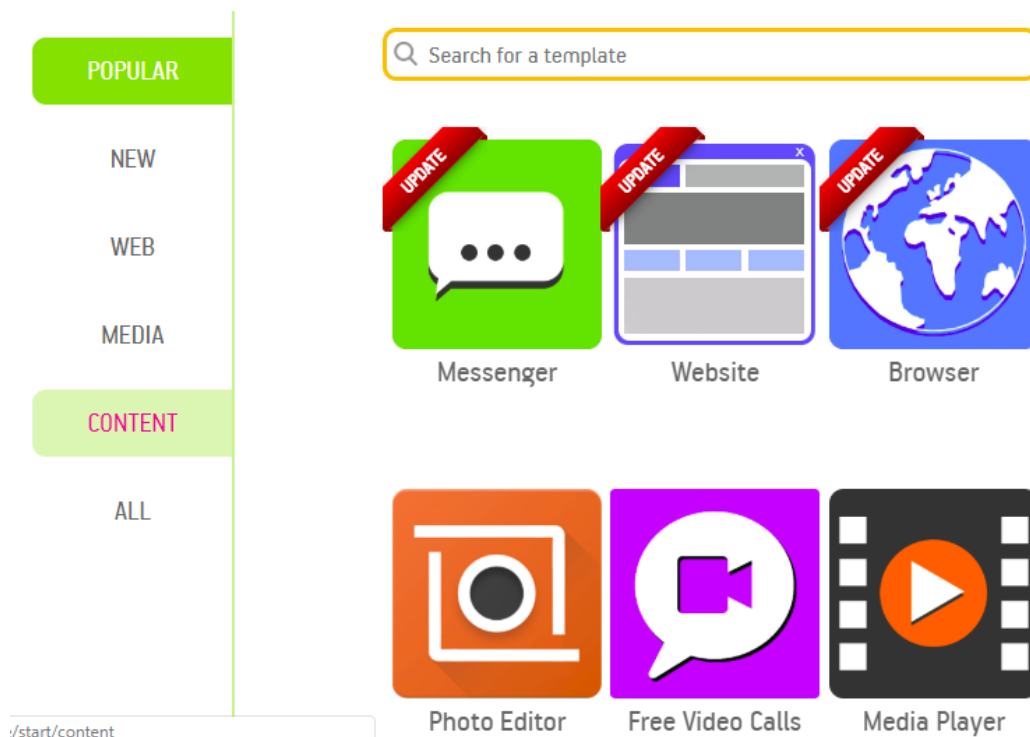
▶ לאחר ההזדהות הראשונית, יפתח לכם החלון הראשי של APPSGEYSER



ההודעה מסבירה שאין לנו
עדיין אף אפליקציה ויש
להתחיל לפתח אותה

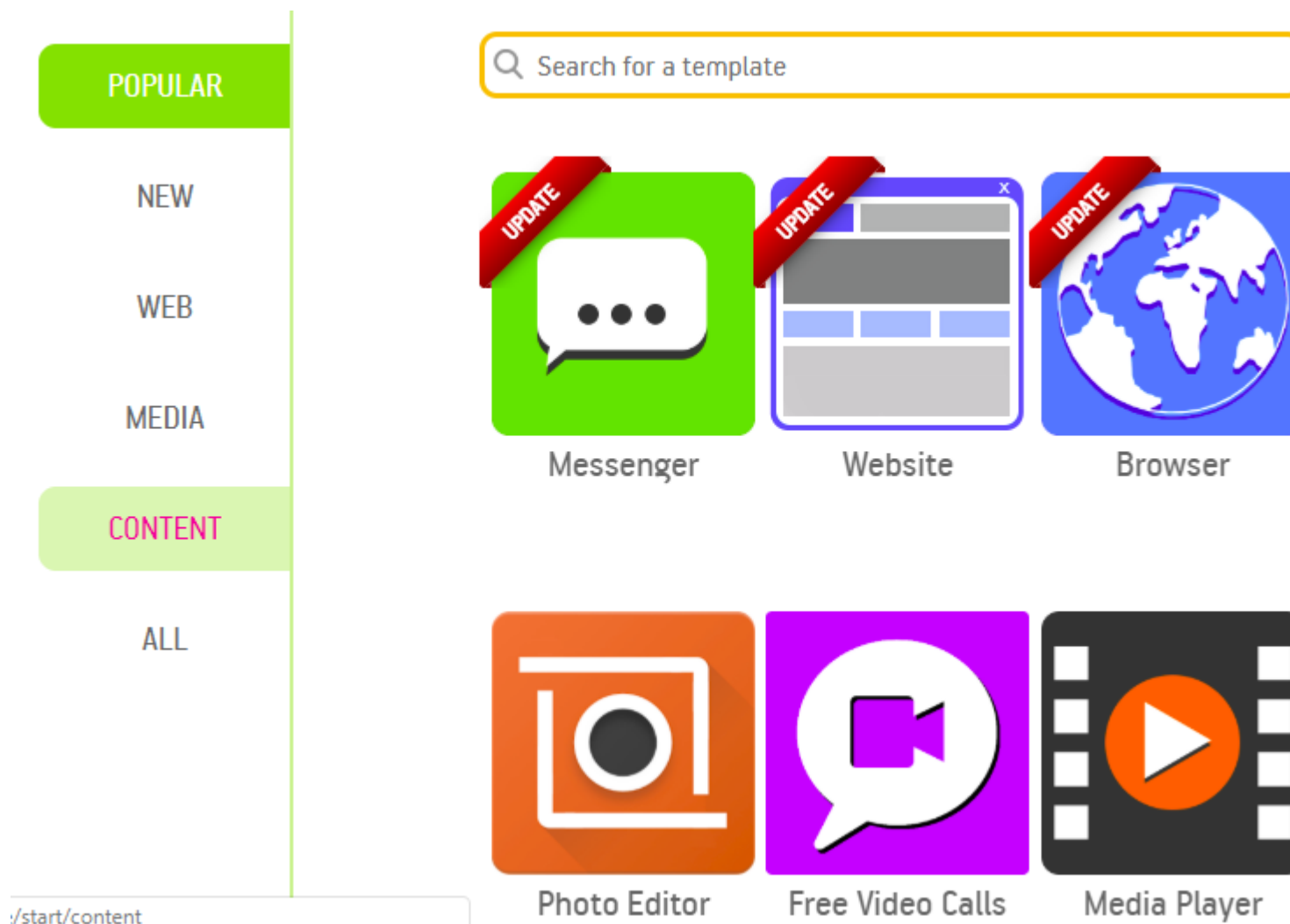
פיתוח אפליקציה ראשונה

- ▶ לאחר שבחרנו לפתח אפליקציה, יפתח עבורינו מסך המאפשר לבחור סוג אפליקציה שנרצה לפתח
- ▶ עלינו להחליט מראש על איזה אפליקציה נתבסס
- ▶ באתר קיימים מאות אלפי סוגי אפליקציות



פיתוח אפליקציה ראשונה

ב-APPSGEYSER קיימים מספר סוגים עיקריים - אתר, משחק או ומדיה



פיתוח אפליקציה ראשונה

- ▶ לצורך הדוגמא נבחר במשחק - מצא סוג AR SHOOTER APP
- ▶ המשחק היפה הזה מאפשר ירי בתמונות באמצעות השימוש המצלמה של המכשיר הסלולארי
- ▶ התמונות והתיאור של המשחק אנחנו יכולים לבחור



Runner game



AR shooter



News Topics

פיתוח אפליקציה ראשונה

- ▶ במסך הבא - יוצג לנו סוג של אשף שאיתו נבצע פיתוח אפליקציה
- ▶ ברגע שמילינו חלק אחד, אפשר לעבור לחלק הבא
- ▶ אשף מבוסס אקורדיון ברגע שממלאים חלק א, נפתח חלק ב - אפשר לעבור בין החלקים

CREATE FIND THE PAIR APP

 Refresh preview

TEMPLATE DESCRIPTION

Create your own **Find the Pair** game!
You can create different levels (e.g. Animals, Flags, Emoji) and change their appearance.

UI assets are from: <http://www.fleamedia.com/category/free-resource>

You can find different free UI assets [here](#)

Fields marked with * symbol are **required**

APP SETTINGS

Having troubles with this form? Follow simple [guide](#).

Click here to see
the preview

פיתוח אפליקציה ראשונה

במסך זה יש לבחור תמונות של האנשים הרעים נגדם תשחקו - המשחק כאמור מציג את התמונות "הרעים" מעל המצלמה

יש ללחוץ על הכפתור ATTACH IMAGE ולבחור תמונות

יש לבחור מתוך התמונות מצורפות

ENEMIES

You can find different enemies packs [here](#) and [here](#)

Add a easy enemy. Killed with 1 shot:

Name: *

Image: * Attach image [Remove image](#)



Select the enemy size: ☐ 1 ☒ 2



פיתוח אפליקציה ראשונה

ניתן גם לבחור את גודל האויב

מתוך מספרים מ-1-5

Add a medium enemy. Killed with 2 shots:

Name: *

איש בינוני

Image: *

Attach image

[Remove image](#)



Select the
enemy size:

- ☐ 1
- ☐ 2
- ☐ 3
- ☒ 4

פיתוח אפליקציה ראשונה

- ▶ בשלב הבא יש לבחור אפשרות המשחק
- ▶ לכל שלב ואפשרות יש לבחור תמונה
- ▶ יש להוסיף גם תאור לכל שלב במשחק

GAME MODES

There are several game modes in the game. You need to specify a name, an icon and a description for each of these game modes.

Tutorial icon
(256x256): *

Attach image

Remove image



First game mode with mostly easy enemies.

Name examples: Scouts first, Warm Up, Hunt, Stroll

Easy level
name: *

BAD MAN

פיתוח אפליקציה ראשונה

במידה ושכחתם להוסיף תאור לשלב מסוים

התוכנה תזכיר לכם לגבי זה

התוכנה תציג לכם את ההודעה - THE FIELD IS REQUIRED - חובה להזין את השדה

Last game mode with all the enemies.
Name examples: Survival, The Final Battle

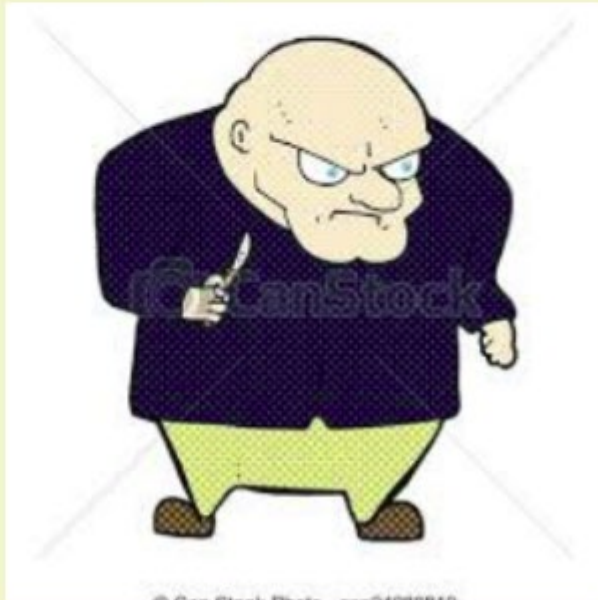
Survival level
name:*

This field is required

Survival level
icon
(256x256):*

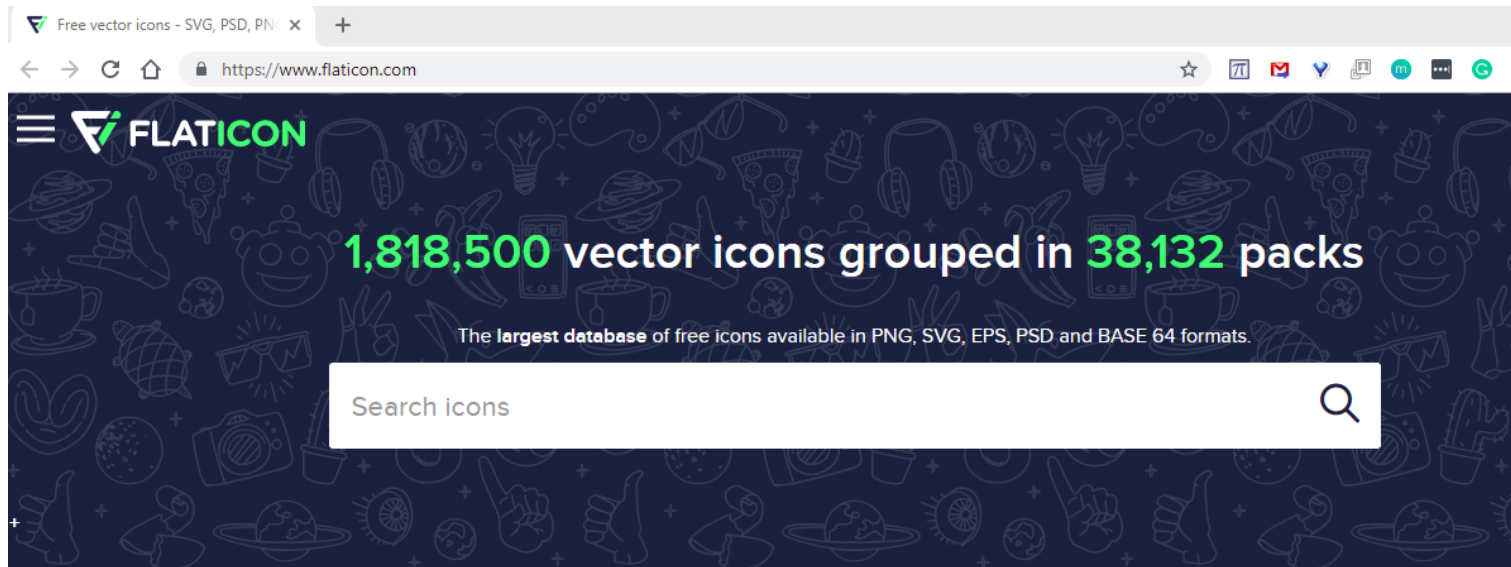
Attach image

[Remove image](#)



פיתוח אפליקציה ראשונה

- במידה וחסר לכם תמונות
- ניתן להיכנס לאתר תמונות ולמצוא תמונות נוספות
- FLATICON = זה האתר המומלץ בAPPGEYSER



Best packs of the month [Browse](#)



פיתוח אפליקציה ראשונה

כאן יש להוסיף את שמה של האפליקציה APP NAME

יש ללחוץ על NEXT

TEMPLATE DESCRIPTION

ENEMIES

GAME MODES

MENU IMAGES

GAME SOUNDS

APP NAME

BADMAN11

NEXT



הסבר APPGEYSER

במידה ואתם מתקשים יש מדריך מפורט שמסביר בשפת אנגלית לגבי בעיות הקיימות

<http://www.appsgeyser.com/testing/>

תאור המשחק APPGEYSER

לבסוף יש לכתוב את התאור של המשחק

DESCRIPTION

תאור המשחק - משחק עוסק במחלמה קשה אותה מנהלים גיבורים הרעים נגד הטובים
אלינו לנצח את האנשים הרעים!!!



NEXT

הזמת אייקון המשחק APPGEYSER

יש לבחור גם את האייקון של המשחק

ICON

☒ Default icon

☐ Custom icon



NEXT

הזמת אייקון המשחק APPGEYSER

אחרי שמילאתם את כל השדות שיש ללחוץ על הכפתור CREATE ליצירת המשחק
בהצלחה!!!

MENU IMAGES

GAME SOUNDS

APP NAME

DESCRIPTION

ICON

CREATE

Fine tune your app after clicking Create App

Click here to see
the preview

Preview may not work for some
templates. Your actual app can look
differently on your Android device.

הזנת פרטים APPGEYSER

ניתן להזין את הטלפון והשם שלכם אבל לא חובה לעשות זאת!!!!

×

Together we can make your app better!

Share your phone number and your name to keep in touch with us and receive critical notifications about your app.

Phone *

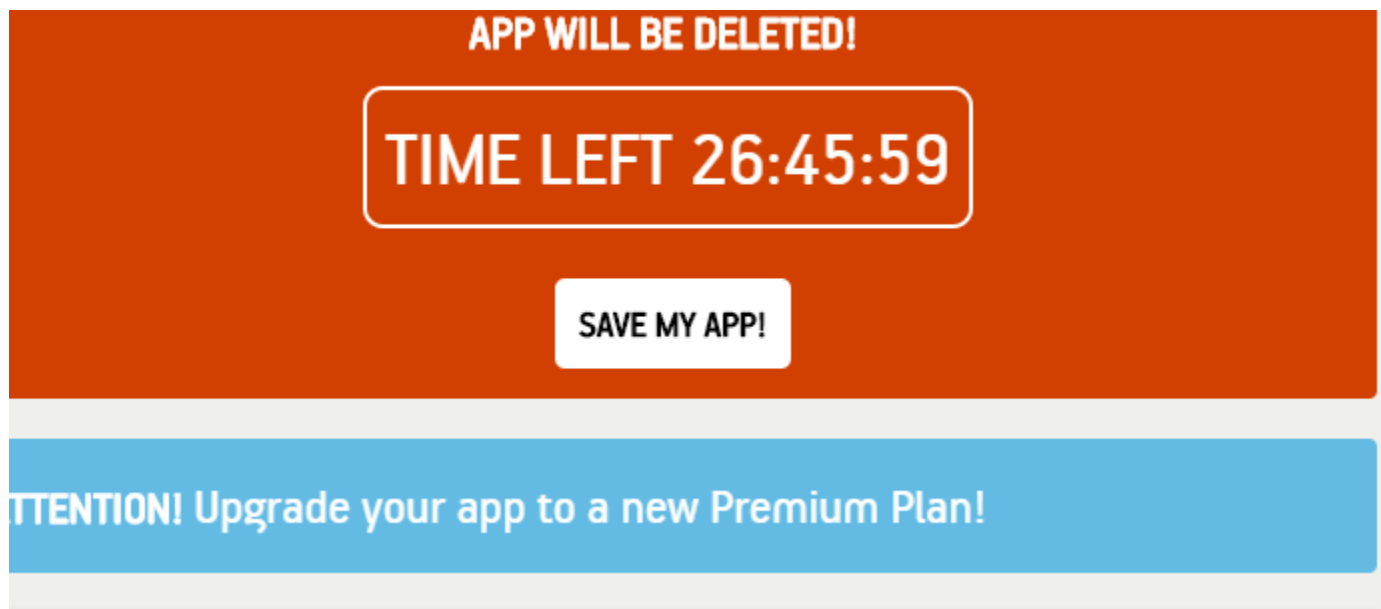
 +972

Name

SUBMIT!

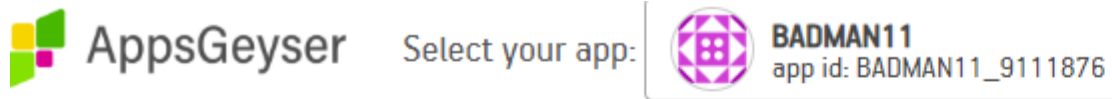
שמירת אפליקציה הראשונה

לאחר יצירת האפליקציה היא תשמר במשך 26 שעות באתר
במידה ולא תשמרו אותה היא תמחק אחרי 26 שעות - !!!



הורדת קובץ הAPK

- בשיעור הקודם דיברנו לגבי קובץ הAPK
- קובץ הAPK מכיל את ההתקנה של האפליקציה שלנו
- לצורך המשחק באפליקציה, נוריד את קובץ הAPK ונשמור אותו במחשב שלנו

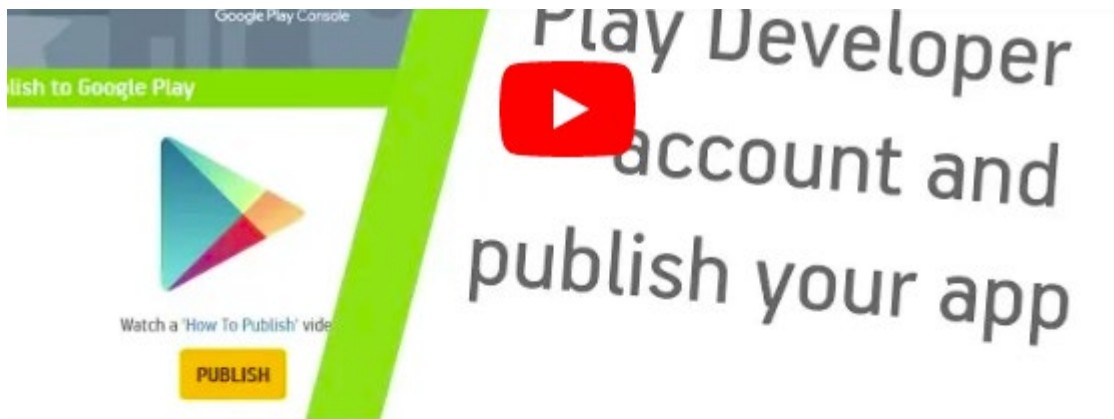


הורדת קובץ APK



הורדת קובץ ה-APK

- ▶ בשיעור הקודם דיברנו לגבי קובץ ה-APK
- ▶ קובץ ה-APK מכיל את ההתקנה של האפליקציה שלנו
- ▶ לצורך משחק באפליקציה, נוריד את קובץ ה-APK ונשמור אותו במחשב שלנו



Publish

[or test your app](#)

App is ready!



הורדת קובץ ה־APK

אחרי שהורדנו את הקובץ ה־APK למחשב, נעלה אותו ול־GOOGLE DRIVE ונשלח לעצמינו שימו לב!!! לא ניתן לשלוח את הקובץ דרך המייל מכיוון שהמייל חוסם את קבצי ה־APK הדרך הנכונה לשלוח קובץ הוא באמצעות ה־GOOGLE DRIVE

Insert files using Google Drive

My Drive

Shared with Me

Recent

Upload

 _BADMAN11_9111876.apk 12.33M

Add more files

Upload

Cancel

Insert as

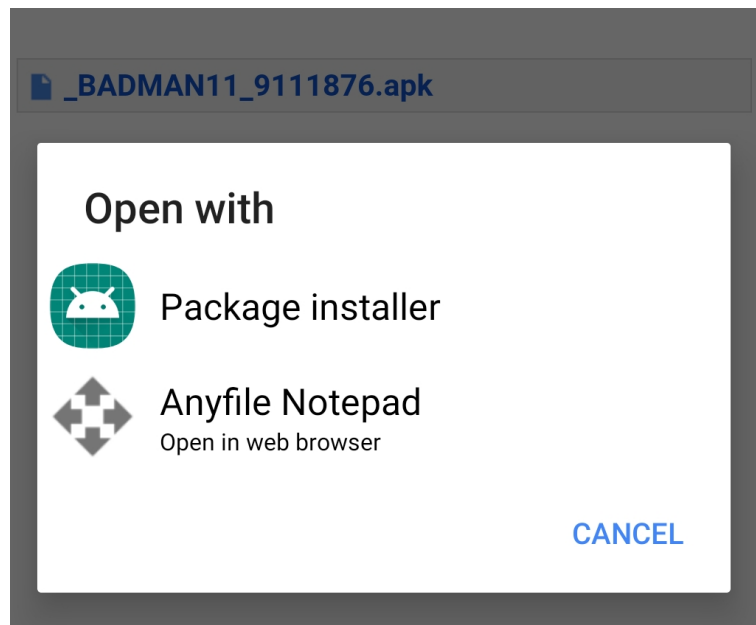
 Drive link

 Attachment

התקנת האפליקציה

יש להוריד את הקובץ מ-GOOGLE DRIVE או מהמייל

יש ללחוץ על הקובץ ולבחור באפשרות התקנה - PACKAGE INSTALLER



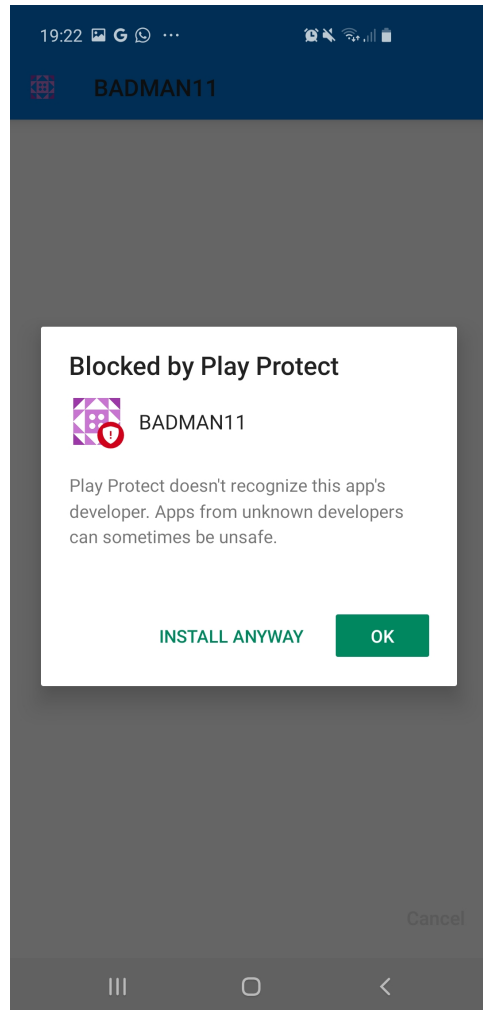
התקנת האפליקציה

אנדרואיד יודא שאכן אתם מעוניינים לבצע את ההתקנה של האפליקציה



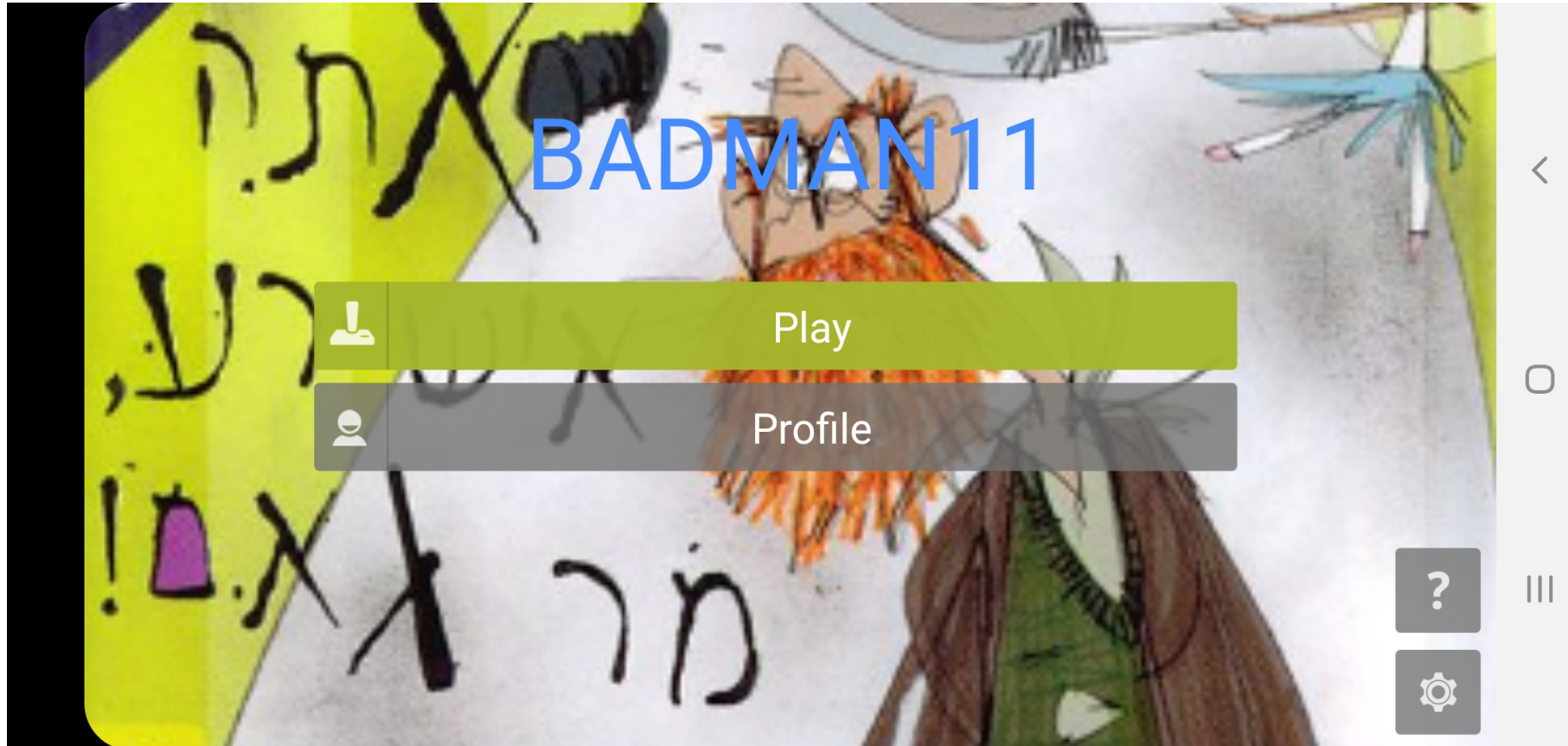
התקנת האפליקציה

יש לאפשר את ההתקנה של האפליקציה



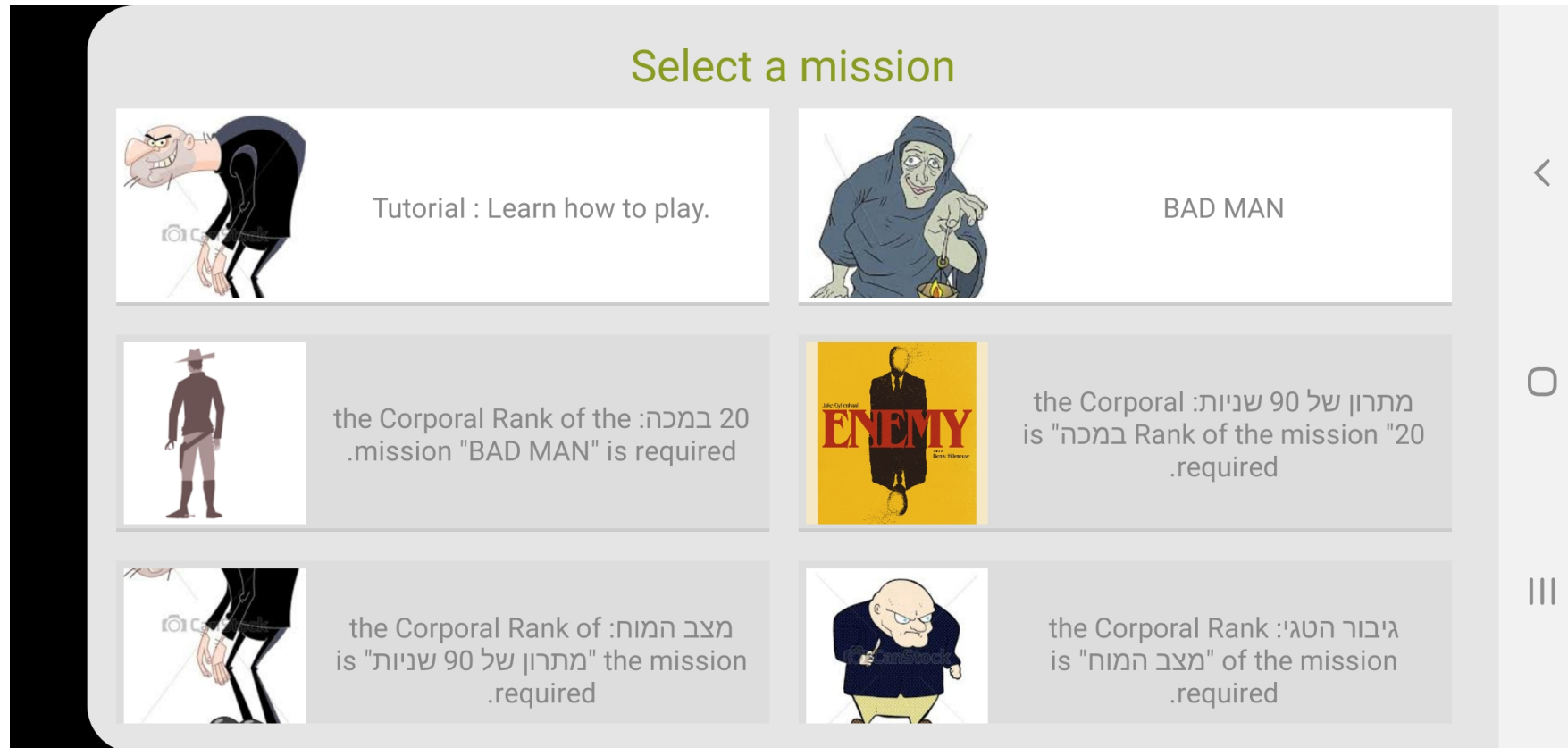
התקנת האפליקציה

מסך ראשי של האפליקציה לאחר ההתקנה משלב את התמונות שהוספנו



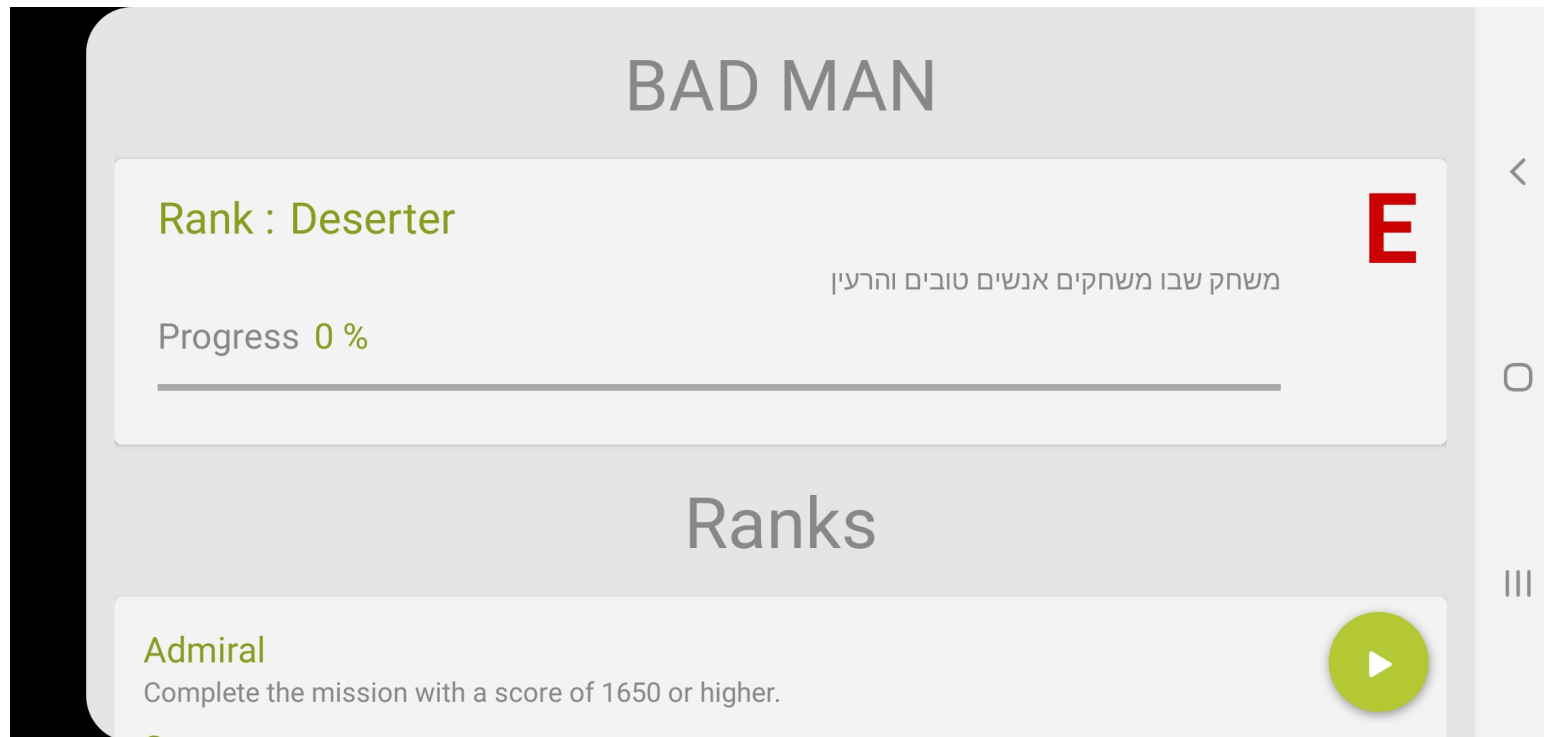
התקנת האפליקציה

גם התמונות וגם הטקסט משולבים בתוך האפליקציה



התקנת האפליקציה

גם ההסברים למשחק - מתוך הטקסט שכתבתנו עבורה



התקנת האפליקציה

גם המשחק עצמו, ירי לכוון התמונות משלב את האפליקציה והתמונות שהעלנו



תרגיל מספר 2

יש למצוא ולהפעיל אפליקציה TIC TAC TOE באמצעות APP GEYSER

יש למצוא ולהפעיל משחק נוסף - לפי בחירתכם