

פיתוח אפליקציות שיעור מספר 1

שיעור 1 - שאלות ותשובות + תרגול

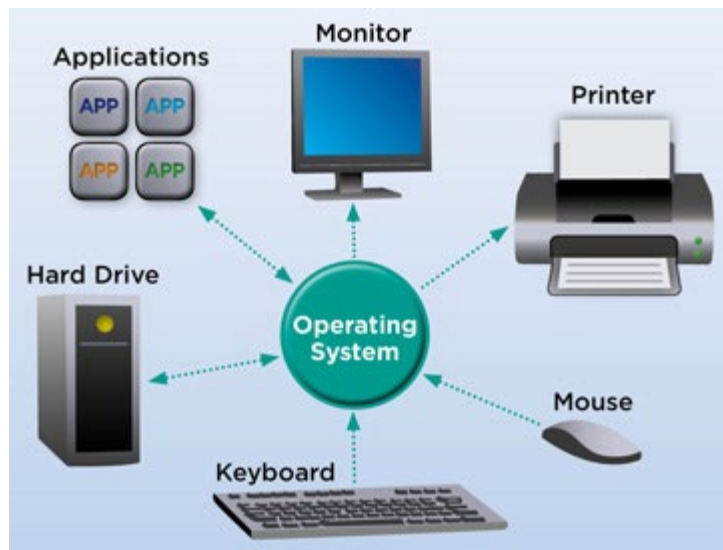
1. היכרות
2. מה זה מערכת ההפעלה?
3. מה זה מערכת ההפעלה אנדרואיד?
4. מה זה אפליקציה?
5. מה עושים במהלך הקורס?
6. מה זה APP INVERTOR ?
7. פיתוח אפליקציה הראשונה - HELLO WORLD

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 ►

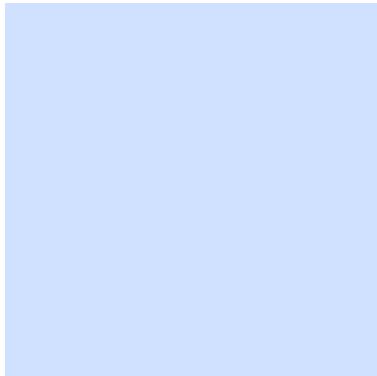
LINUX ►

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



מערכת הפעלה LINUX

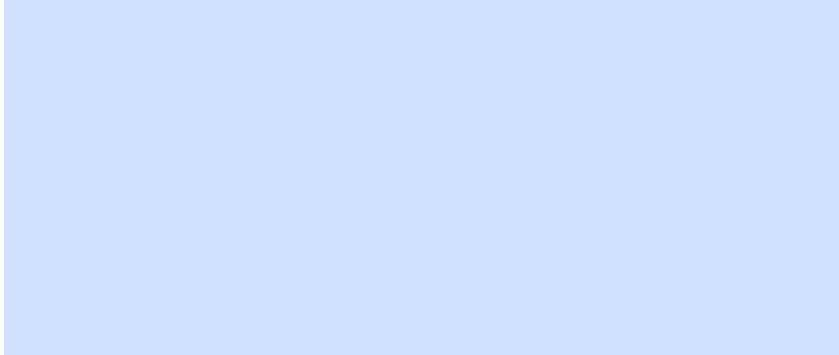
- ▶ מה אתם יודעים על LINUX?
- ▶ מערכת הפעלה פתוחה - מבוססת על OPEN SOURCE - קוד פתוח
- ▶ ניתן להוריד בחינם
- ▶ ניתנת לפיתוח לתכניתנים בכל העולם
- ▶ הרבה ארגונים בעולם הורידו את הקוד של לינוקס בחינם והרחיבו ושינו אותו והוציאו LINUXים משלהם - נקרא "הפצה של לינוקס"
- ▶ בעצם אנחנו משתמשים עכשיו בלינוקסים 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

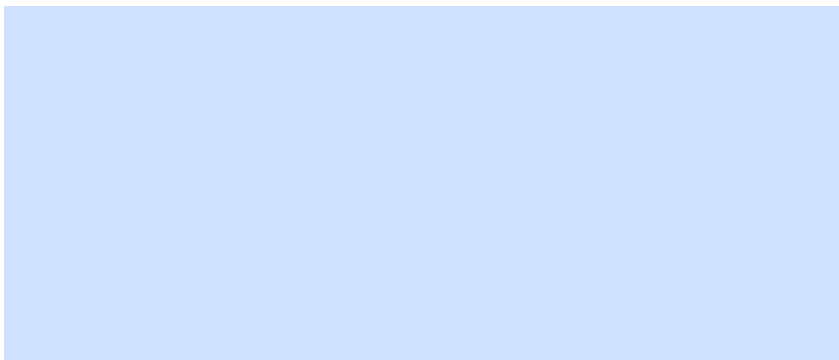
- ▶ המערכת מגדירה הרשאות עבור כל הקבצים ביישום כך שרק מזהה המשתמש שהוקצה ליישום זה יוכל לגשת אליהם.
- ▶ לכל תהליך יש מחשב וירטואלי משלו (VM), כך קוד אפליקציה פועל בבידוד מיישומים אחרים.
- ▶ כברירת מחדל, כל יישום פועל בתהליך לינוקס משלה. מערכת אנדרואיד מתחיל את התהליך כאשר כל הרכיבים של האפליקציה צריך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

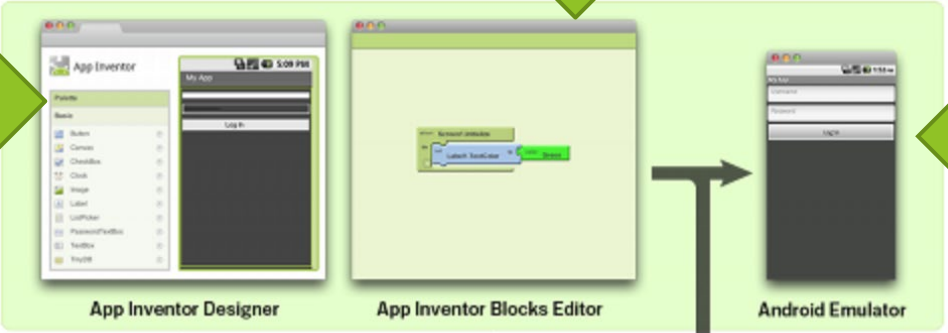
- ▶ אפליקציות חינוכיות - כולל וידאו, תמונות, טקסט לדיבור
- ▶ אפליקציות מבוססות מיקום
- ▶ משחקים
- ▶ יישומים מותאמי אינטרנט
- ▶ אפליקציות אישיות

שרתי
APP
INVERTOR
בענן



בניית לוגיקה
באשף לוגיקה

בניית מסכים
באמצעות
אשף
DESIGNER



התקנת
אפליקציה על
סימולטור

התקנת
אפליקציה על
מכשיר
הטלפון



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

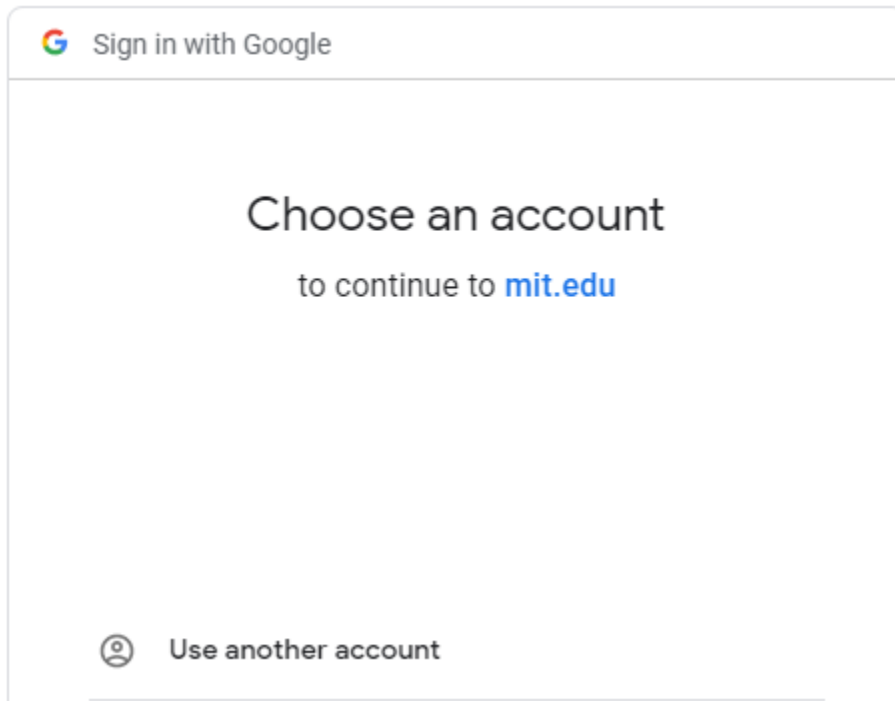
יש לוודא שהמחשב מחובר לאינטרנט ►

יש ליצור חשבון GMAIL ►

יש להתחבר ל ►

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

לאחר ההתחברות, תנובת להזנת חשבון GMAIL ►



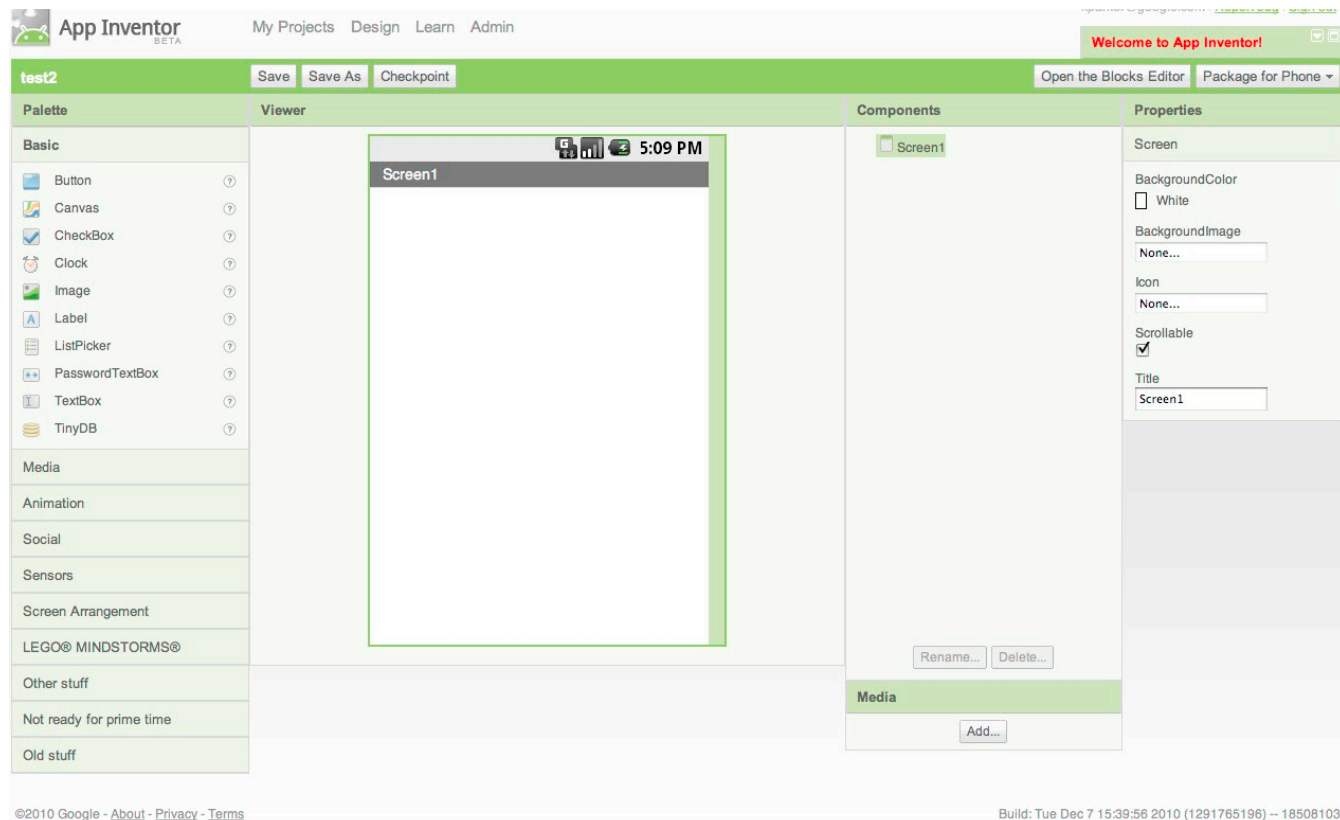
הפעלה הראשונה של APP INVERTOR

- ▶ 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

The screenshot displays the App Inventor interface, divided into two main sections: the Palette on the left and the Viewer on the right.

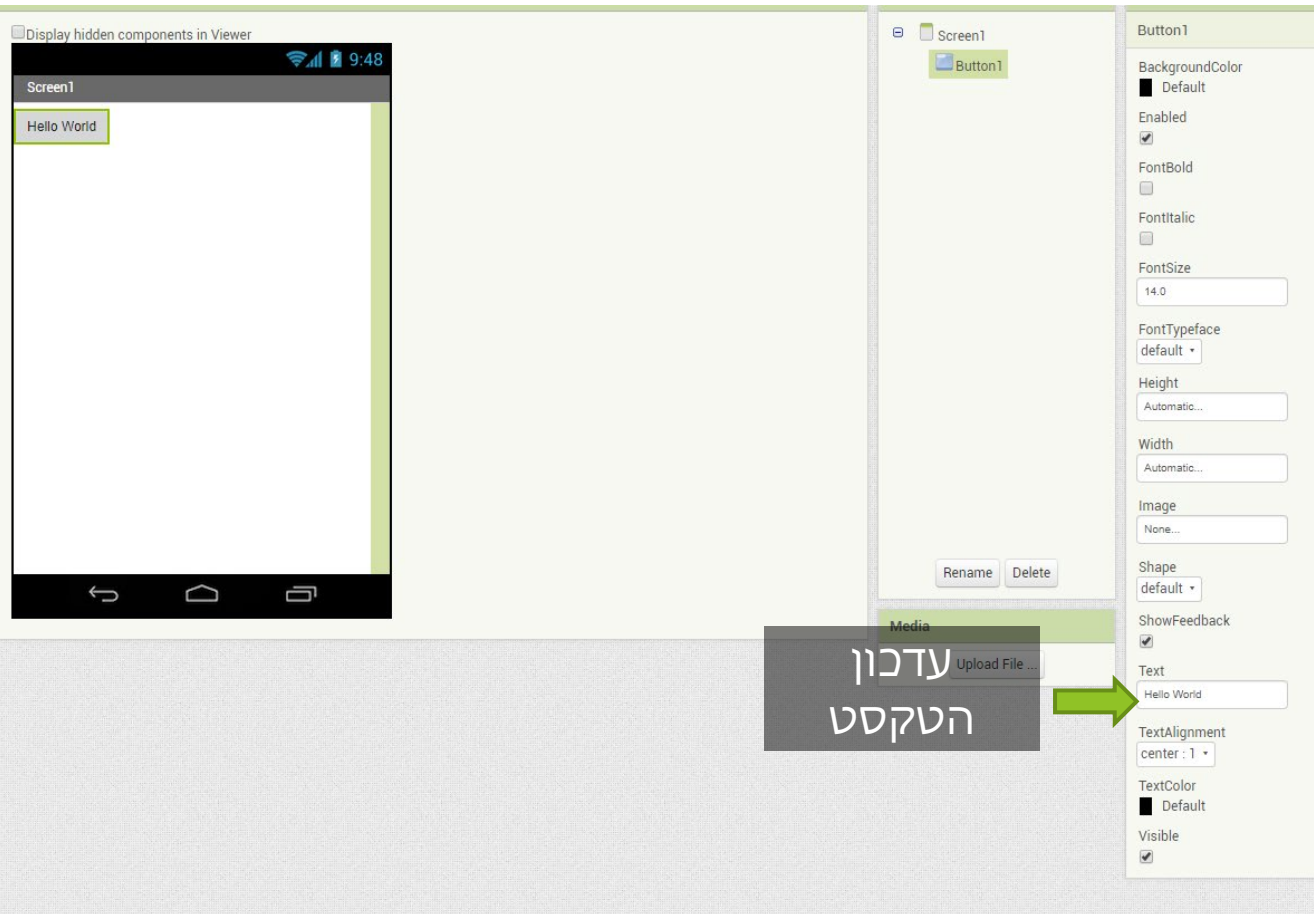
Palette: This section contains a list of user interface components under the heading "User Interface". The components listed are: Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, and Spinner. Each component has a small icon and a help icon (a question mark in a circle). The "Button" component is currently selected and highlighted in green.

Viewer: This section shows a preview of the application screen. At the top, there is a status bar with icons for Wi-Fi, signal strength, battery, and the time 9:48. Below the status bar, the screen is titled "Screen1". A text box containing the text "Text for Button1" is visible on the screen, and it is highlighted with a green border. A large green arrow points from the "Button" component in the Palette to this text box in the Viewer, indicating that the text is the label for the button.

At the top right of the Viewer area, there is a checkbox labeled "Display hidden components in Viewer" which is currently unchecked.

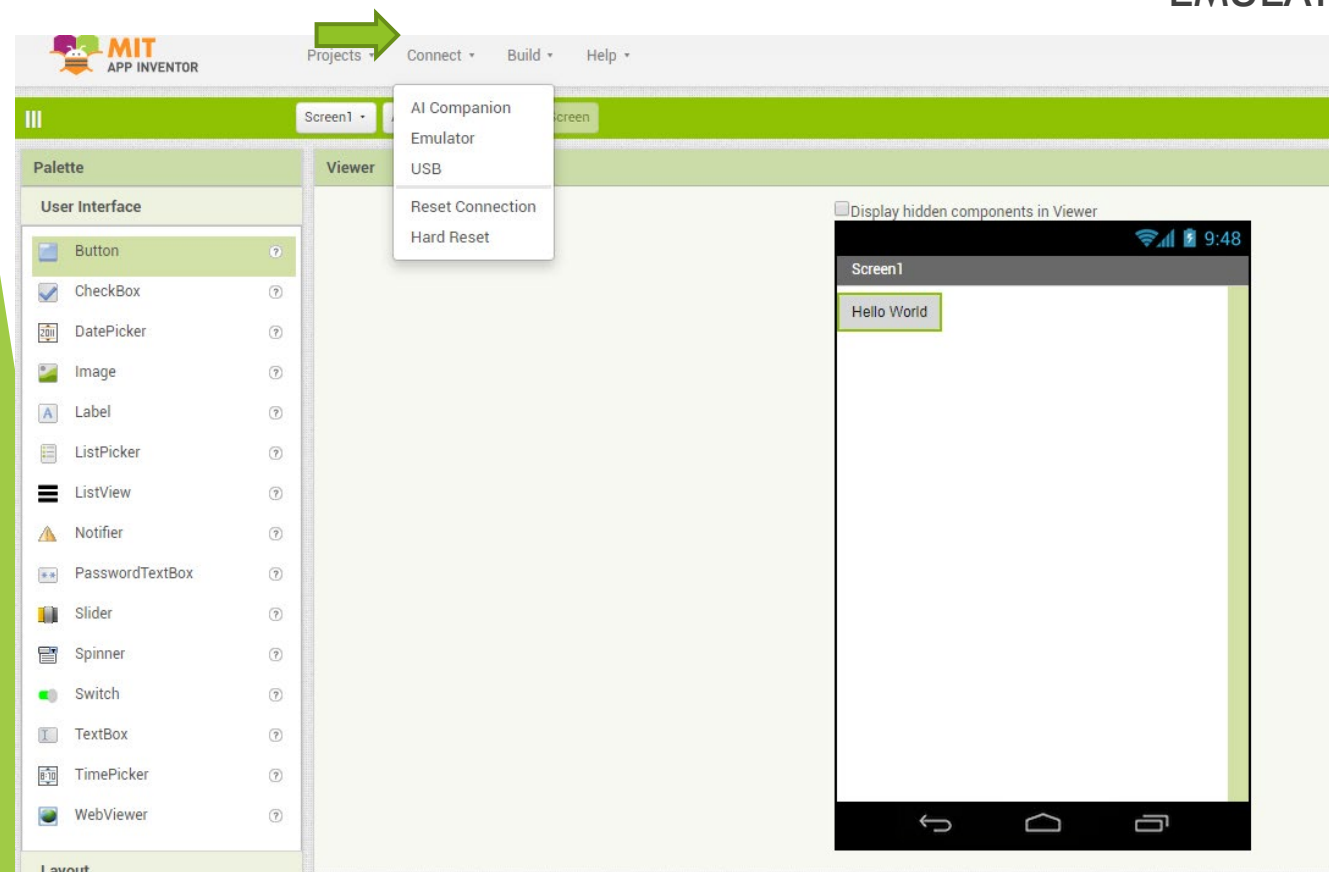
הפעלה הראשונה של APP INVERTOR

- ▶ 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



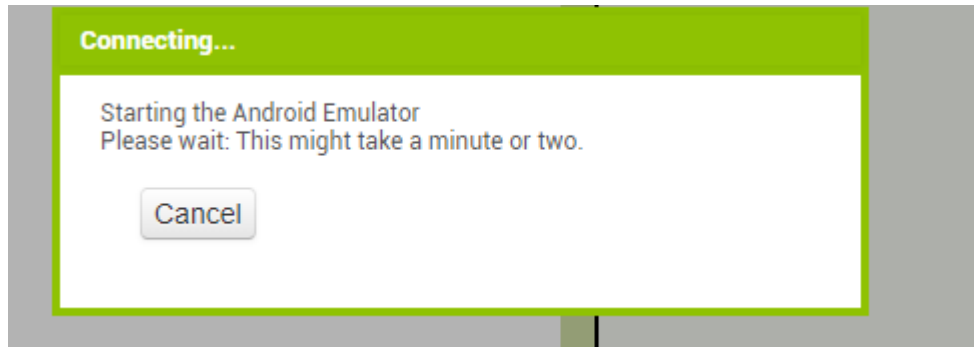
הפעלה ראשונה של APP INVERTOR

- ▶ 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

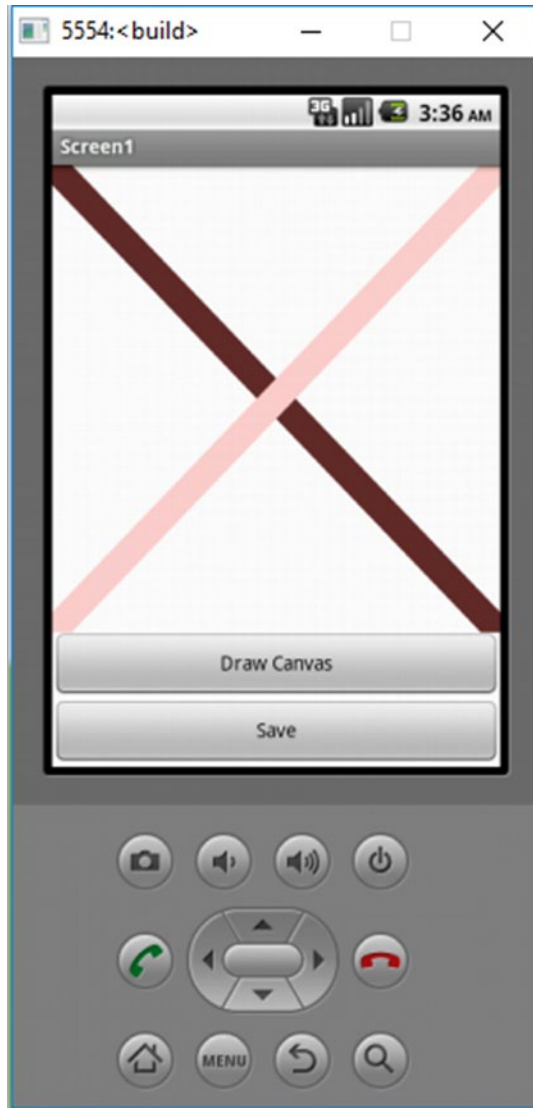


הפעלה ראשונה של APP INVERTOR

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!!!) ►

The screenshot displays the App Inventor web interface. At the top, the title bar reads "Lesson1" and includes buttons for "Screen1", "Add Screen ...", and "Remove Screen". The main workspace is divided into several panels:

- Palette:** A list of user interface components such as Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. The "CheckBox" component is currently selected.
- Viewer:** A central area showing a mobile device simulation. The screen displays a button labeled "Talk2Me". Above the viewer, there is a checkbox labeled "Display hidden components in Viewer".
- Components:** A tree view on the right side of the workspace showing the hierarchy of components: "Screen1" containing "Button1" and "TextToSpeech1".
- Properties:** A panel on the far right showing the properties for the selected "TextToSpeech1" component. The visible properties include "Country" (set to "Default"), "Language" (set to "Default"), "Pitch" (set to "1.0"), and "SpeechRate" (set to "1.0").
- Media:** A panel at the bottom right with an "Upload File ..." button.

פיתוח האפליקציה המדברת

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

The screenshot displays the App Inventor web interface. At the top, the title bar reads "Lesson1" and includes buttons for "Screen1", "Add Screen...", and "Remove Screen". On the right side of the top bar are "Designer" and "Blocks" tabs. The interface is divided into several panels:

- Palette:** A list of user interface components such as Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. The "CheckBox" component is currently selected.
- Viewer:** A central area showing a mobile device simulation. The screen displays "Screen1" with a button labeled "Talk2Me". A checkbox above the viewer is labeled "Display hidden components in Viewer".
- Components:** A tree view showing the hierarchy of components on the screen: "Screen1" containing "Button1" and "TextToSpeech1".
- Properties:** A panel for the selected "TextToSpeech1" component, showing settings for "Country" (Default), "Language" (Default), "Pitch" (1.0), and "SpeechRate" (1.0).
- Media:** A section at the bottom right with an "Upload File ..." button.

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

The screenshot displays the App Inventor web interface. At the top, the title bar reads "Lesson1" and includes buttons for "Screen1", "Add Screen...", and "Remove Screen". The interface is divided into several panels:

- Palette:** A list of UI components under the "User Interface" section, including Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. The "CheckBox" component is currently selected.
- Viewer:** A central area showing a mobile device simulation. The screen displays "Screen1" with a button labeled "Talk2Me". A checkbox above the viewer is labeled "Display hidden components in Viewer".
- Components:** A tree view on the right showing the hierarchy of components on the screen: "Screen1" containing "Button1" and "TextToSpeech1".
- Properties:** A panel on the far right showing the properties for the selected "TextToSpeech1" component, including "Country" (Default), "Language" (Default), "Pitch" (1.0), and "SpeechRate" (1.0).
- Media:** A section at the bottom right with an "Upload File ..." button.

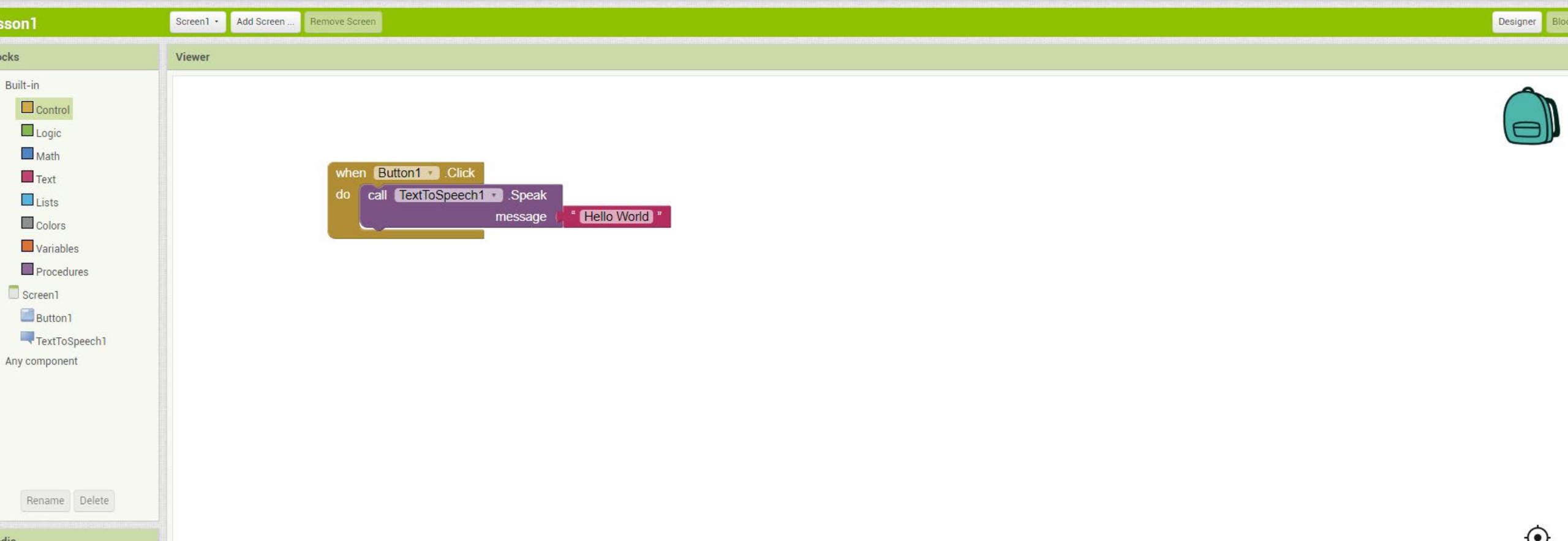
פיתוח אפליקציה מדברת

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

The screenshot displays a visual programming interface with a green header bar. The header contains the text 'Lesson1' on the left, 'Screen1', 'Add Screen ...', and 'Remove Screen' in the center, and 'Designer' and 'Block' on the right. On the left side, there is a 'Blocks' panel with a 'Built-in' section containing categories: Control (orange), Logic (green), Math (blue), Text (red), Lists (light blue), Colors (grey), Variables (orange), Procedures (purple), and Screen1 (white). Under 'Screen1', there are sub-items: Button1 (blue) and TextToSpeech1 (light blue). Below these are 'Any component' and 'Rename' and 'Delete' buttons. The main 'Viewer' area shows a script starting with a 'when Button1.Click' block (orange), followed by a 'do' block (purple) containing a 'call TextToSpeech1.Speak' block (purple), which in turn contains a 'message' block (red) with the text 'Hello World'.

פיתוח אפליקציה מדברת

הפעלת את התוכנית בדרך הרגילה (על ידי אמולטור) ►



The screenshot displays the Scratch IDE interface. At the top, the title bar shows "Scratch" and "Screen1". Below the title bar, there are buttons for "Add Screen...", "Remove Screen", "Designer", and "Block palette". On the left side, the "Blocks" palette is visible, showing categories like Control, Logic, Math, Text, Lists, Colors, Variables, Procedures, and Screen1. The main workspace, labeled "Viewer", contains a script for a button click event. The script starts with a "when Button1 .Click" block, followed by a "do" block containing a "call TextToSpeech1 .Speak" block and a "message" block with the text "Hello World".

```
when Button1 .Click
do
  call TextToSpeech1 .Speak
  message "Hello World"
```

פתיחת אפליקציה קיימת - MAGIC

טענו אפליקציה MAGIC ▶

אפליקציה עובדת עם תמונות שהכנו מראש ▶

The screenshot displays the MAGIC IDE interface. On the left, a mobile app preview window shows a black top hat with a brown band, a 'restart' button, and a status bar at the top with the time 9:48. Below the preview, a 'Non-visible components' section lists 'Sound1' and 'AccelerometerSensor1'. On the right, the 'Components' panel shows a tree view with 'Screen1' containing 'Button1', 'Restart', 'Sound1', and 'AccelerometerSensor1'. The 'Properties' panel for 'Button1' shows settings like 'BackgroundColor', 'Enabled', 'FontSize', and 'Image'. The 'Media' panel at the bottom lists files like 'TaDasound.mp3', 'duckhat1.jpg', and 'hat.jpg'. A green arrow points from the 'Media' panel to a green box containing the Hebrew word 'תמונות' (Images).

פתיחת אפליקציה קיימת - MAGIC

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

```
when Button1 .Click
do
  set Button1 . Image to "rabbit.jpg"
  call Sound1 .Play

when Restart .Click
do
  set Button1 . Image to "hat.jpg"

when AccelerometerSensor1 .Shaking
do
  set Button1 . Image to pick a random item list get global Ducklist
  call Sound1 .Play

initialize global Ducklist to
  make a list
  "duckhat1.jpg"
  "duckhat2.jpg"
  "duckhat3.jpg"
```

פיתוח אפליקציות

שיעור מספר 2

- .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

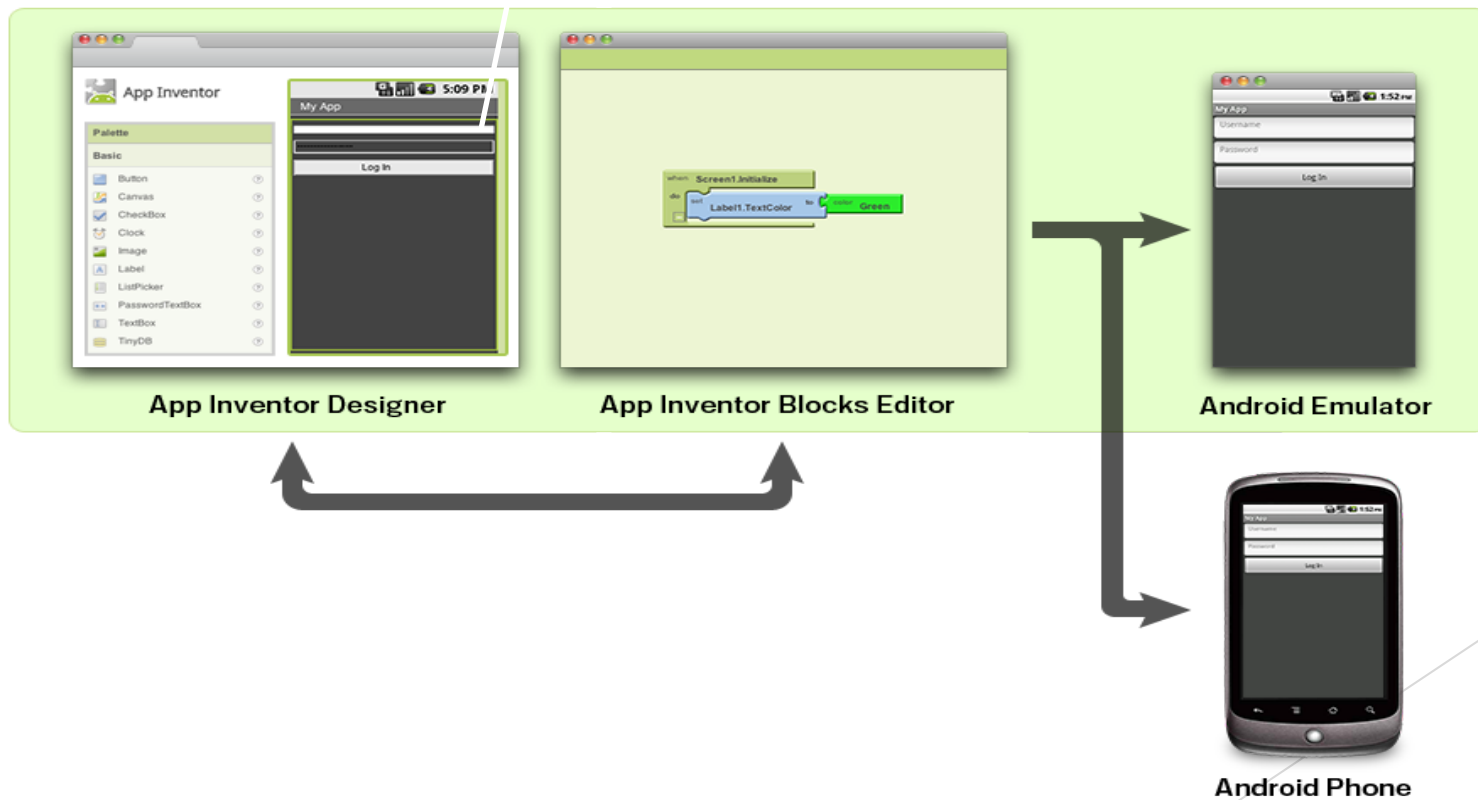
שרתי ה APP INVERTOR מאחסנים את העבודות שלך
וגם עוזרים לך לעקוב אחר הפרויקטים שלך.



Google App Inventor Servers



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



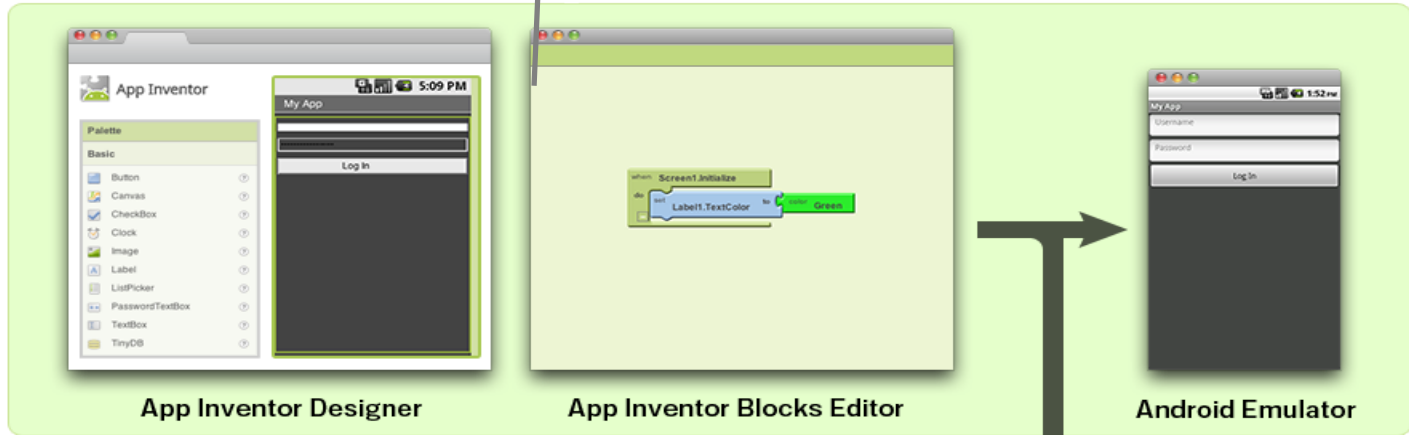
ממשיכים להכיר APP INVERTOR



Google App Inventor Servers



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.



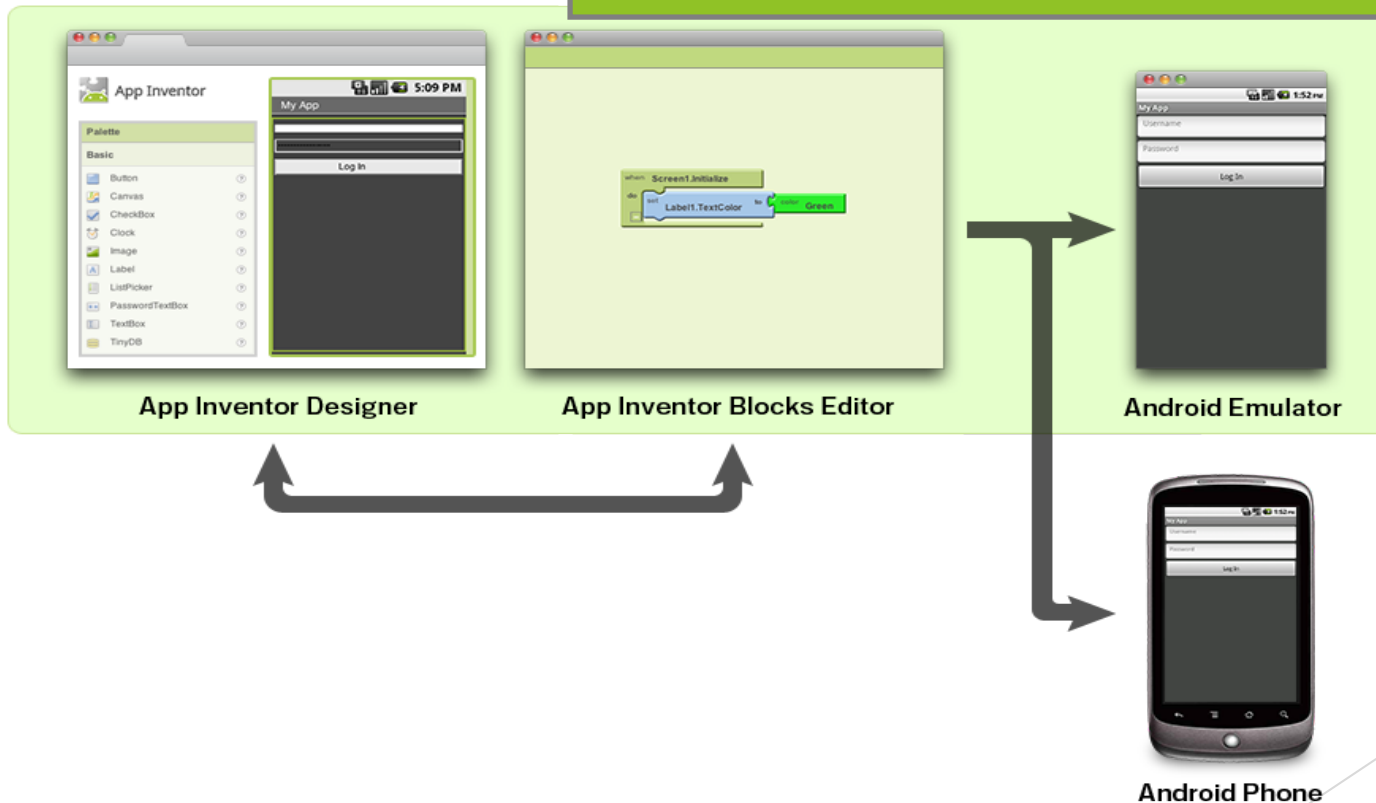
Android Phone



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.

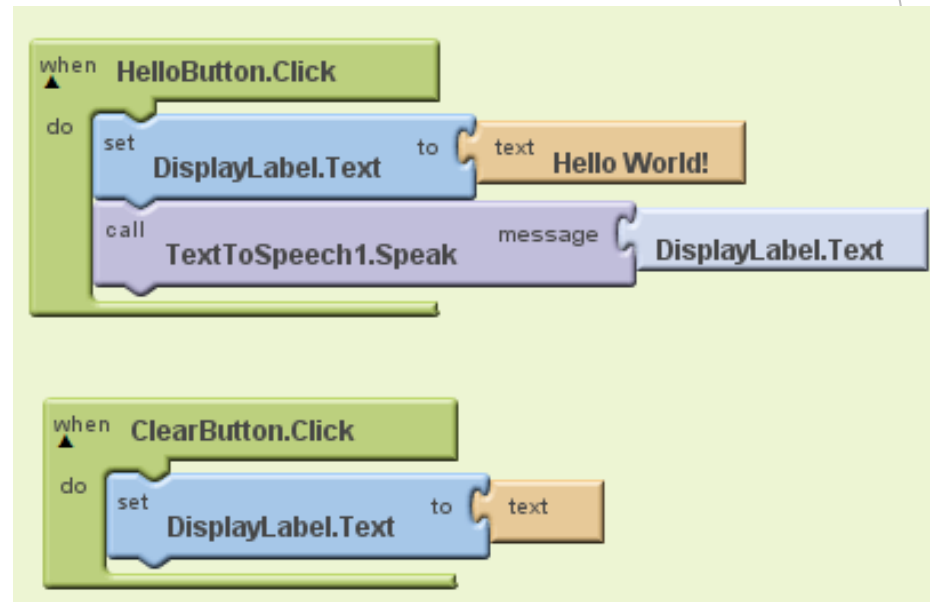


ההשוואה בין Java ל Applinventor

Java Code

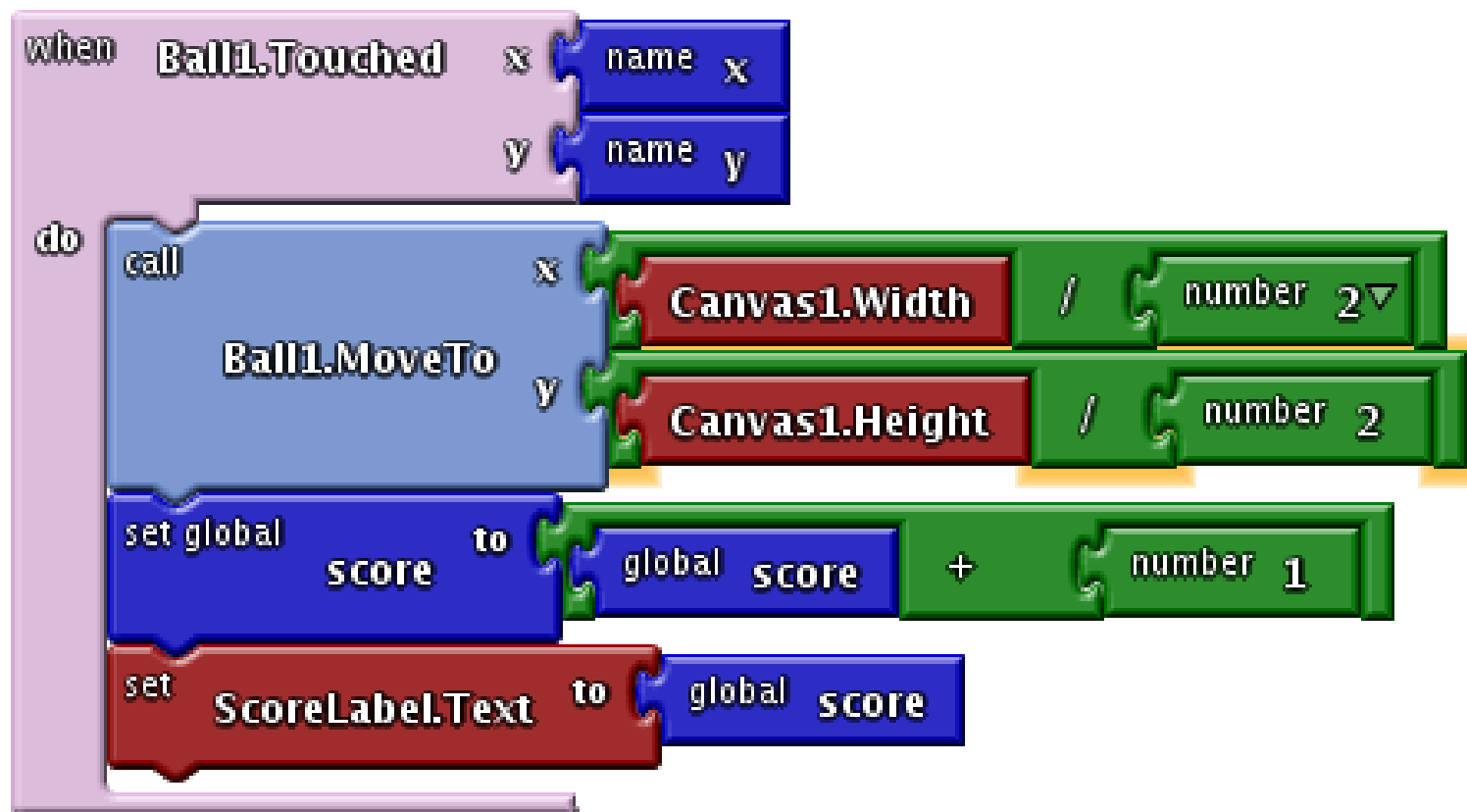
```
public class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

AppInventor

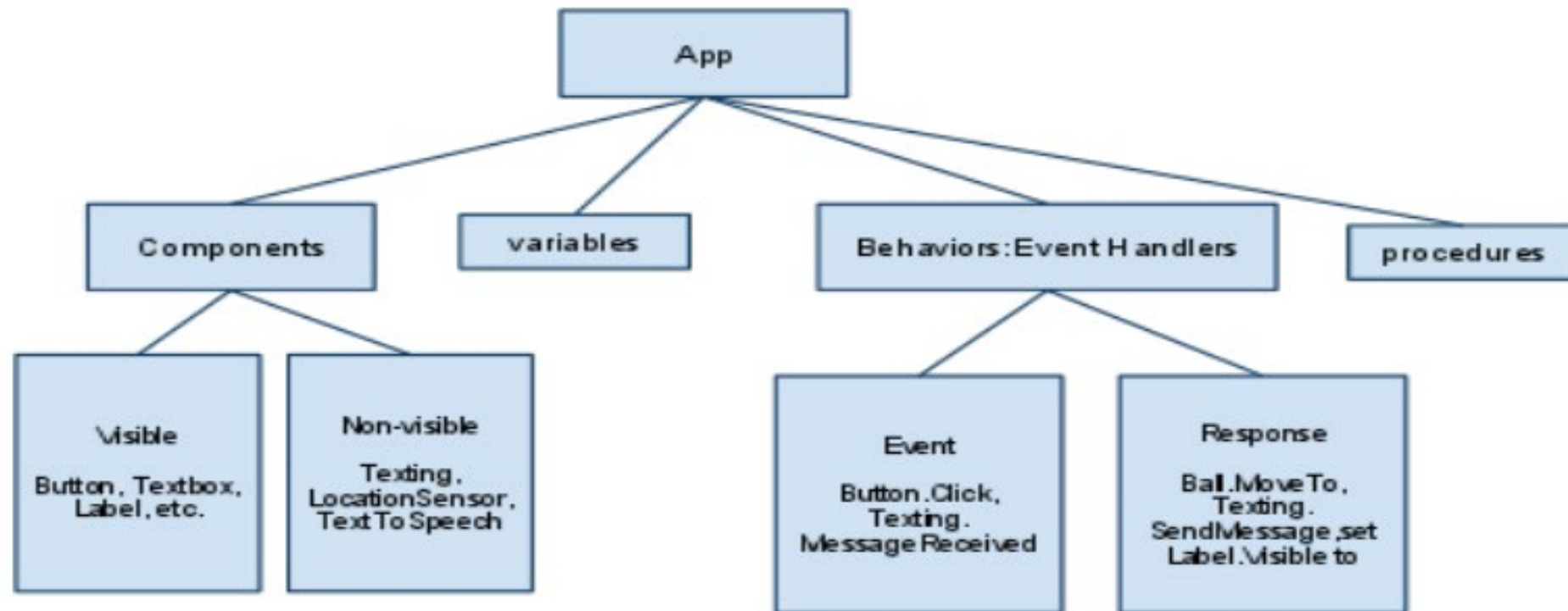


בלוקים

נחשו מה עושים הבלוקים הללו ?



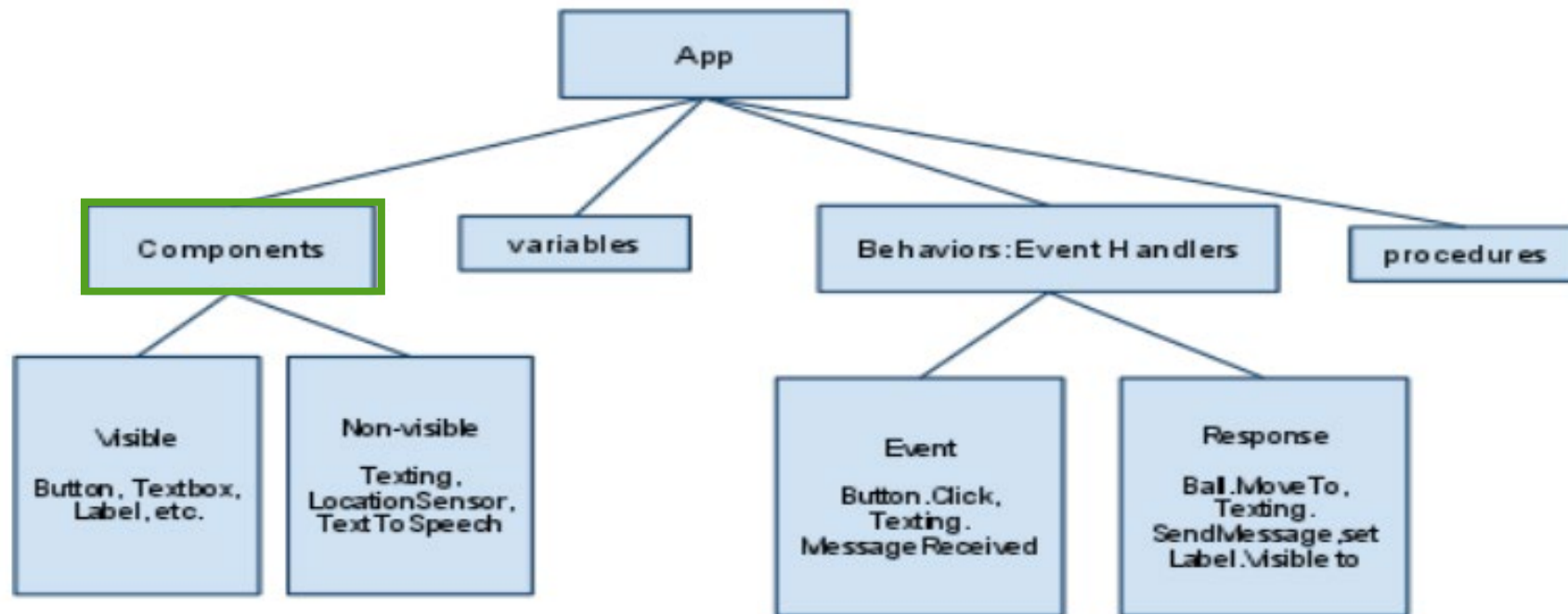
ארכיטקטורת התוכנית - ממה מורכבת האפליקציה??



רכיבים - COMPONENTS - הקומפוננטות

רכיבים הם אובייקטים או רכיבים המשמשים ליצירת יישום.

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



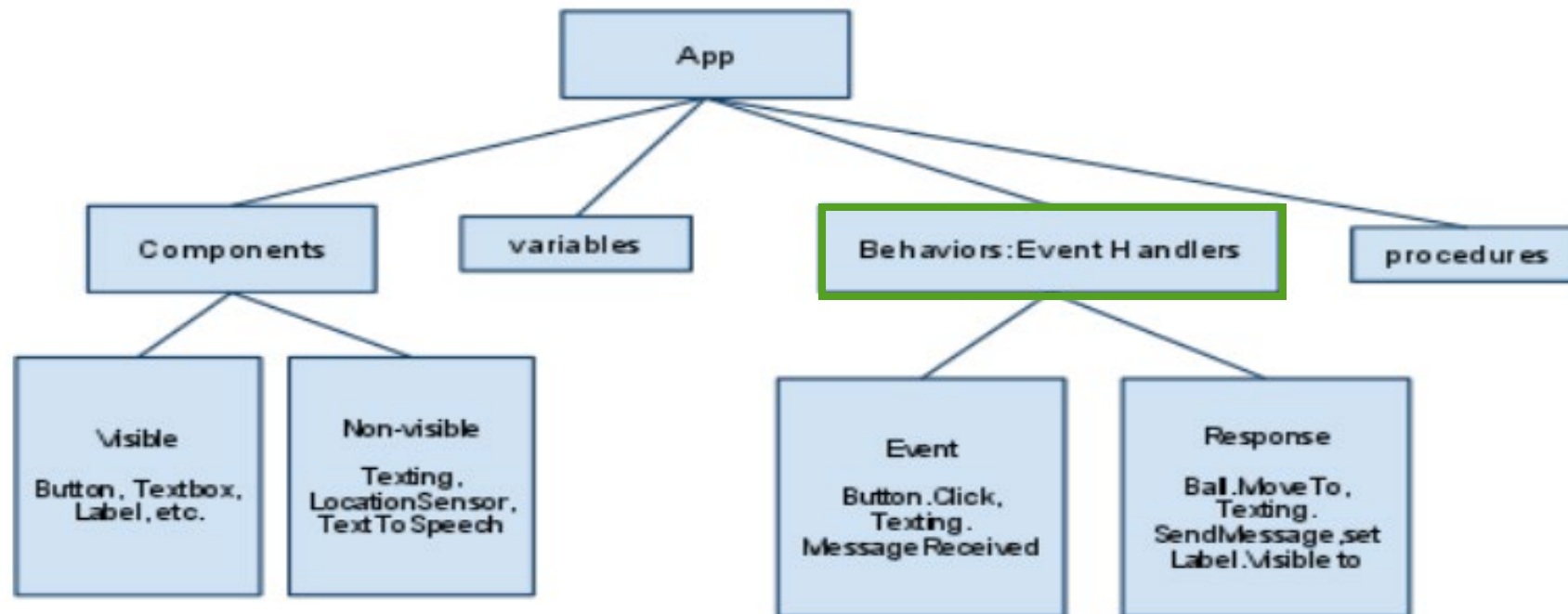
ארכיטקטורת התוכנה

- ▶ 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...

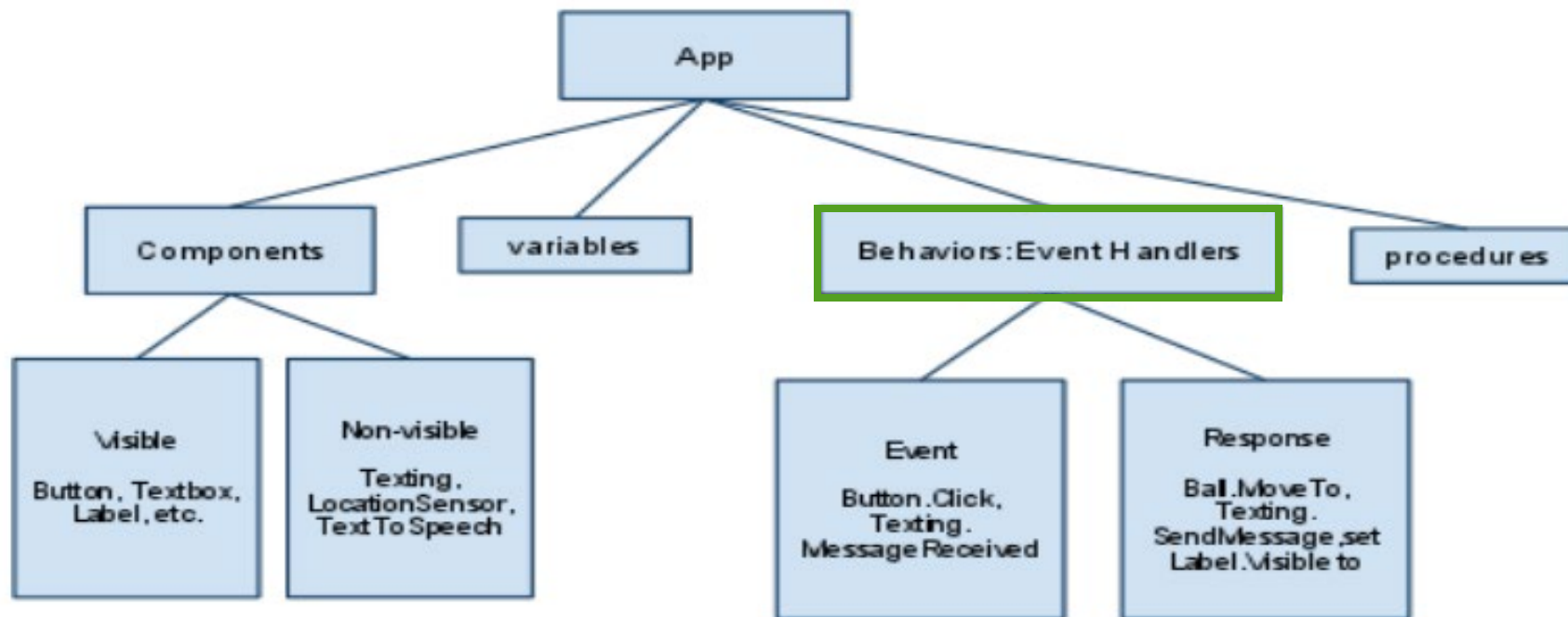
התנהגות - טיפול באירוע - Event handler

▶ 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 ??

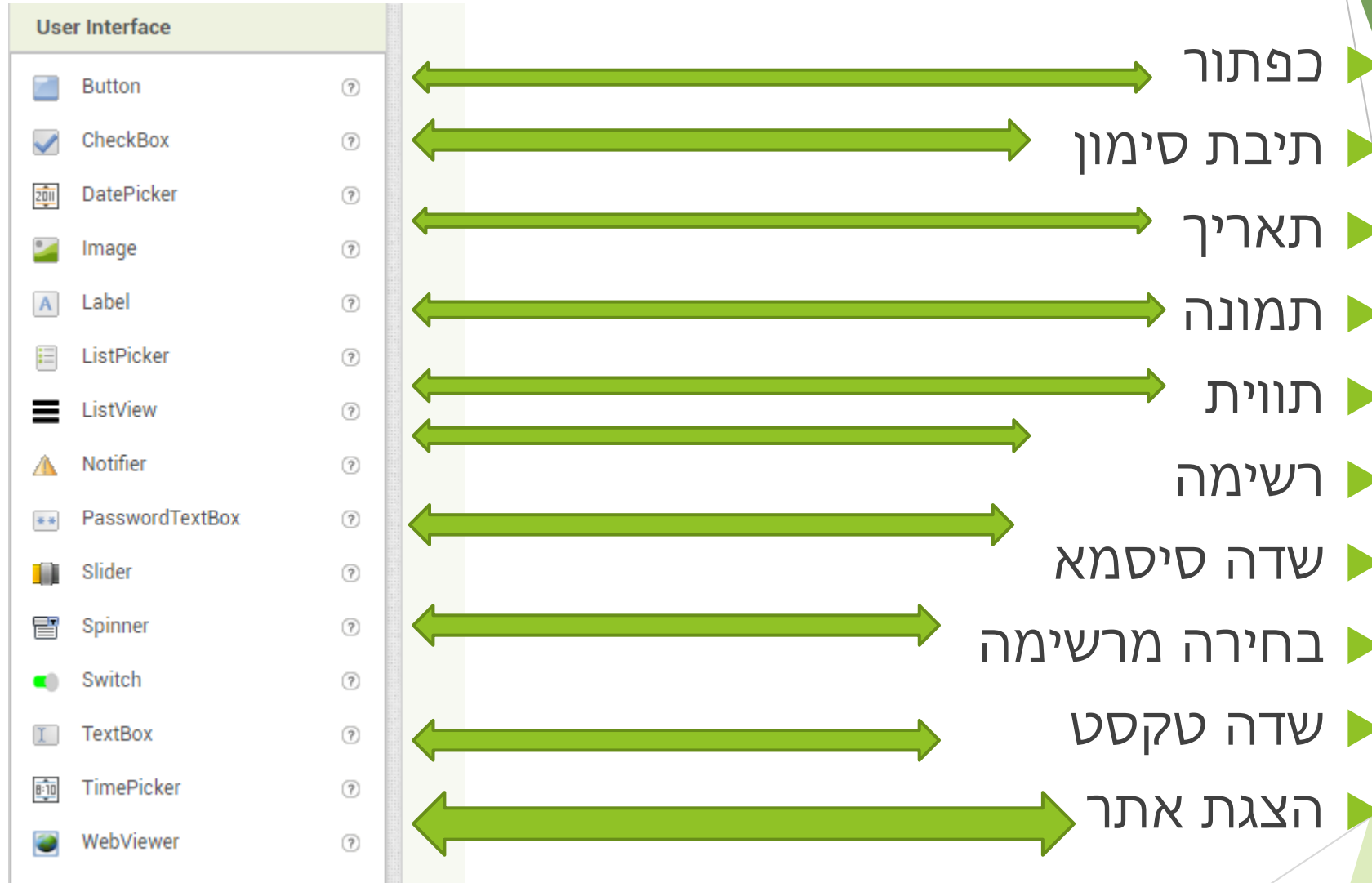
- ▶ אין הקלדה של קוד, אין שגיאות תחביר.
- ▶ אירועים ברמה הראשונה
- ▶ כמו להרכיב פאזל (רק להתאים כמה חתיכות)
- ▶ התכנות הוא ברמה פשוטה - צוות של גוגל הכניס בו הרבה עבודה
- ▶ מאוד ברור ו פחות מופשט

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 ב-APP 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 callout boxes with arrows point from Hebrew text to specific properties:

- קביעת צבע רקע של הכפתור** (Setting the background color of the button) points to **BackgroundColor** (Default).
- קביעת דגש לאותיות בכפתור** (Setting bold for letters on the button) points to **FontBold**.
- קביעת גודל הטקסט** (Setting text size) points to **FontSize** (14.0).
- קביעת גובה הכפתור** (Setting button height) points to **Height** (Automatic...).
- קביעת נטייה לטקסט** (Setting text slant) points to **FontItalic**.
- קביעת פונט לכפתור** (Setting font for the button) points to **FontTypeface** (default).
- קביעת רוחב הכפתור** (Setting button width) points to **Width** (Automatic...).
- ניתן להציג תמונה מעל הכפתור** (Can display an image above the button) points to **Image** (hat.jpg...).

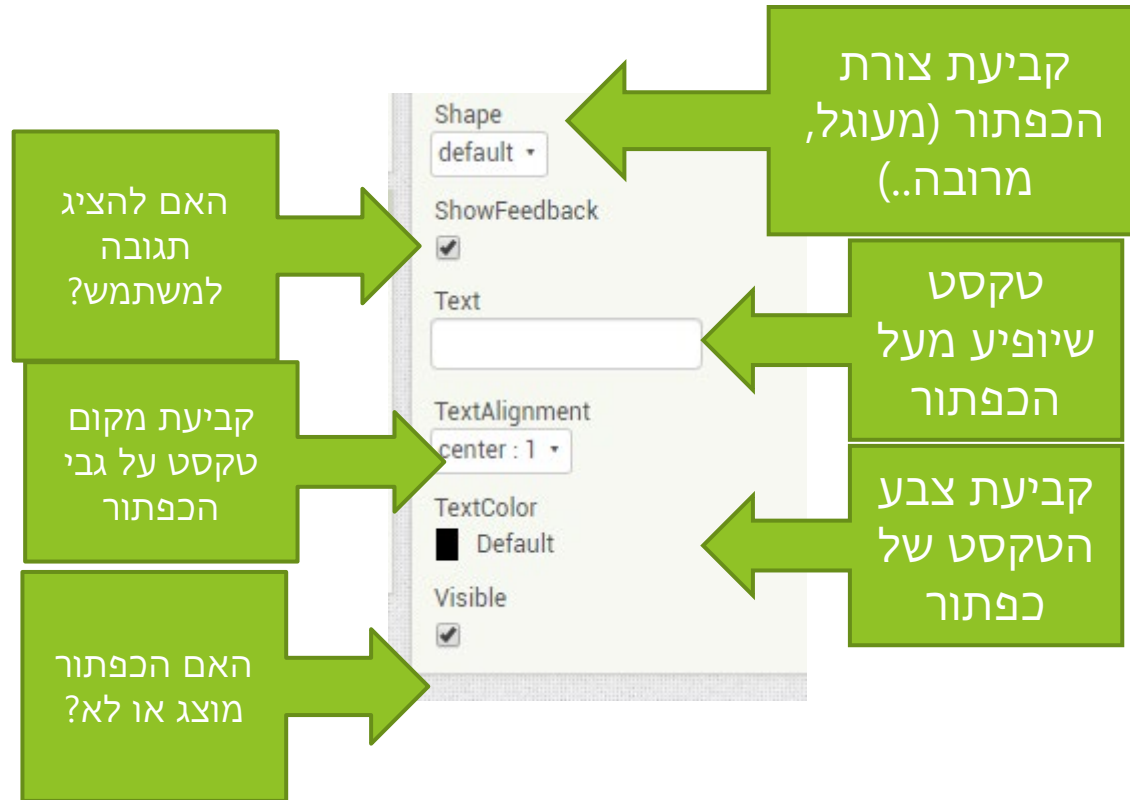
Additional annotations on the left side of the 'Components' panel:

- זמינות לחיצה של הכפתור (האם ניתן או לא ללחוץ על הכפתור)** (Button clickability (can you or not click on the button)) points to the **Enabled** checkbox.
- קביעת צבע של הכפתור** (Setting button color) points to the **BackgroundColor** property.
- קביעת גודל הכתב** (Setting font size) points to the **FontSize** property.
- קביעת נטייה לטקסט** (Setting text slant) points to the **FontItalic** property.
- קביעת פונט לכפתור** (Setting font for the button) points to the **FontTypeface** property.
- קביעת רוחב הכפתור** (Setting button width) points to the **Width** property.

Buttons for 'Rename' and 'Delete' are visible at the bottom of the 'Components' panel. The 'Media' panel at the bottom left shows 'TaDasound.mp3'.

תכונות הכפתור

המשך התכונות...

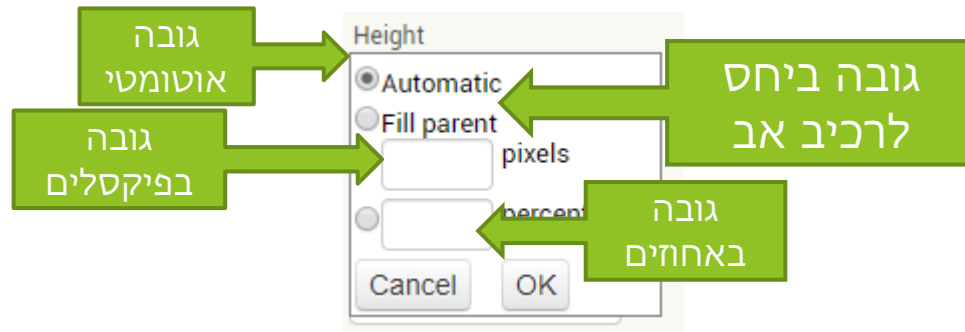


גובה ורוחב של הכפתור | ולא רק כפתור...)

- ▶ 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!!)

The screenshot displays the Android Studio IDE. On the left is a virtual device emulator showing a screen with a grey rectangular button containing the text "Hello World". The status bar at the top of the emulator shows the time as 9:48 and various system icons. In the center is the "Screen1" component tree, with "Button1" selected. On the right is the "Button1" properties panel. The "Height" property is set to "Fill parent...", and a large green double-headed arrow points from this property to the button in the emulator. Other visible properties include BackgroundColor (Default), Enabled (checked), FontBold (unchecked), FontItalic (unchecked), FontSize (14.0), FontTypeface (default), Image (None...), Shape (default), ShowFeedback (checked), and Text (Hello World). At the bottom, there is a "Media" section with an "Upload File ..." button.

הסבר לגדלים

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

The image shows a screenshot of a mobile application development environment. On the left, a preview window displays a mobile screen with a status bar at the top showing the time 9:48 and various icons. Below the status bar, a grey rectangular box contains the text "Hello World". A green arrow points from the text box in the preview to the properties panel on the right. The properties panel is titled "Button1" and lists various attributes: BackgroundColor (Default), Enabled (checked), FontBold (unchecked), FontItalic (unchecked), FontSize (14.0), FontTypeface (default), Height (50 percent...), Width (Automatic...), Image (None...), and Shape (default). At the bottom of the properties panel, there are "Rename" and "Delete" buttons. The overall interface is light green and white.

- ▶ 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 a visual programming environment with two main panels: 'Blocks' on the left and 'Viewer' on the right.

Blocks Panel: Contains a 'Built-in' category with various block types: Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. A red oval highlights the 'Screen1' and 'Button1' components under the 'Screen1' sub-category. Below the list are 'Rename' and 'Delete' buttons. A 'Media' section at the bottom has an 'Upload File ...' button.

Viewer Panel: Shows a vertical stack of event blocks for 'Button1':

- when Button1 .Click do
- when Button1 .GotFocus do
- when Button1 .LongClick do
- when Button1 .LostFocus do
- when Button1 .TouchDown do
- when Button1 .TouchUp do

Below the event blocks are property blocks for 'Button1':

- Button1 . BackgroundColor
- set Button1 . BackgroundColor to
- Button1 . Enabled
- set Button1 . Enabled to
- Button1 . FontBold
- set Button1 . FontBold to
- Button1 . FontItalic
- 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 ה-BLOCKS



BUTTON EVENTS אירועים הכפתור

The screenshot shows a visual programming environment with two main panels: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' panel is divided into 'Built-in' and 'Media' sections. Under 'Built-in', there are categories like Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. A red oval highlights the 'Screen1' and 'Button1' components. Below the 'Built-in' section are 'Rename' and 'Delete' buttons. The 'Media' section has an 'Upload File ...' button. The 'Viewer' panel displays a vertical stack of event blocks for 'Button1'. The top block is 'when Button1 Click', which is highlighted with a red oval. Below it are blocks for '.GotFocus', '.LongClick', '.LostFocus', '.TouchDown', and '.TouchUp'. At the bottom, there are several 'set' blocks for properties: 'BackgroundColor', 'Enabled', 'FontBold', and 'FontItalic'.

- ▶ 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

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

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

איך משנים ערך

מחזיר ערך של
צבע הכפתור

משנה ערך של
צבע הכפתור



שינוי ערך מתבצע עם פעולות SET

לאחר בחלק של BLOCKS לאחר

בחירת כפתור מופיעים בנוסף לאירועים גם תיבות בירוק

התיבות הללו מתחלקות ל 2 חלקים GET ו SET

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

Button1 . BackgroundColor

set Button1 . BackgroundColor to

Button1 . Enabled

set Button1 . Enabled to

Button1 . FontBold

set Button1 . FontBold to

Button1 . FontItalic

set Button1 . FontItalic to

Button1 . FontSize

set Button1 . FontSize to

Button1 . Height

set Button1 . Height to

set Button1 . HeightPercent to

Button1 . Image

set Button1 . Image to

Blocks

Viewer

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

Screen1

- Button1

Any component

Media

Upload File ...

make color

make a list

255

0

0

split color

0

Show Warnings

when Button1 . Click

do set Button1 . BackgroundColor to

- ▶ 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??

Android, Where's My Car? 13:23 67%

Android, Where's My Car?

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

“Where is my car” - ממשק משתמש



“Where is my car” - בלוקים

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

```
when LocationSensor1 .LocationChanged
  latitude longitude altitude
do
  set CurrentAddressDataLabel .Text to LocationSensor1 .CurrentAddress
  set CurrentLatLabel .Text to get latitude
  set CurrentLongLabel .Text to get longitude
  set RememberButton .Enabled to true
```

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.

```
when RememberButton .Click
do
  set RememberedAddressDataLabel .Text to LocationSensor1 .CurrentAddress
  set RememberedLatLabel .Text to LocationSensor1 .Latitude
  set RememberedLongLabel .Text to LocationSensor1 .Longitude
  call TinyDB1 .StoreValue
    tag "address"
    valueToStore LocationSensor1 .CurrentAddress
  call TinyDB1 .StoreValue
    tag "lat"
    valueToStore LocationSensor1 .Latitude
  call TinyDB1 .StoreValue
    tag "long"
    valueToStore LocationSensor1 .Longitude
  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.

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

```
when DirectionsButton .Click
do
  set ActivityStarter1 .DataUri to
    join
      "http://maps.google.com/maps?saddr="
      CurrentLatLabel .Text
      " "
      CurrentLongLabel .Text
      "&daddr="
      RememberedLatLabel .Text
      " "
      RememberedLongLabel .Text
  call ActivityStarter1 .StartActivity
```

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.

```
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
```

אירוע שינוי מיקום

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

```
? when LocationSensor1 .LocationChanged
  latitude longitude altitude
do
  set CurrentAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set CurrentLatLabel . Text to get latitude
  set CurrentLongLabel . Text to get longitude
  ? set RememberButton . Enabled to true
```

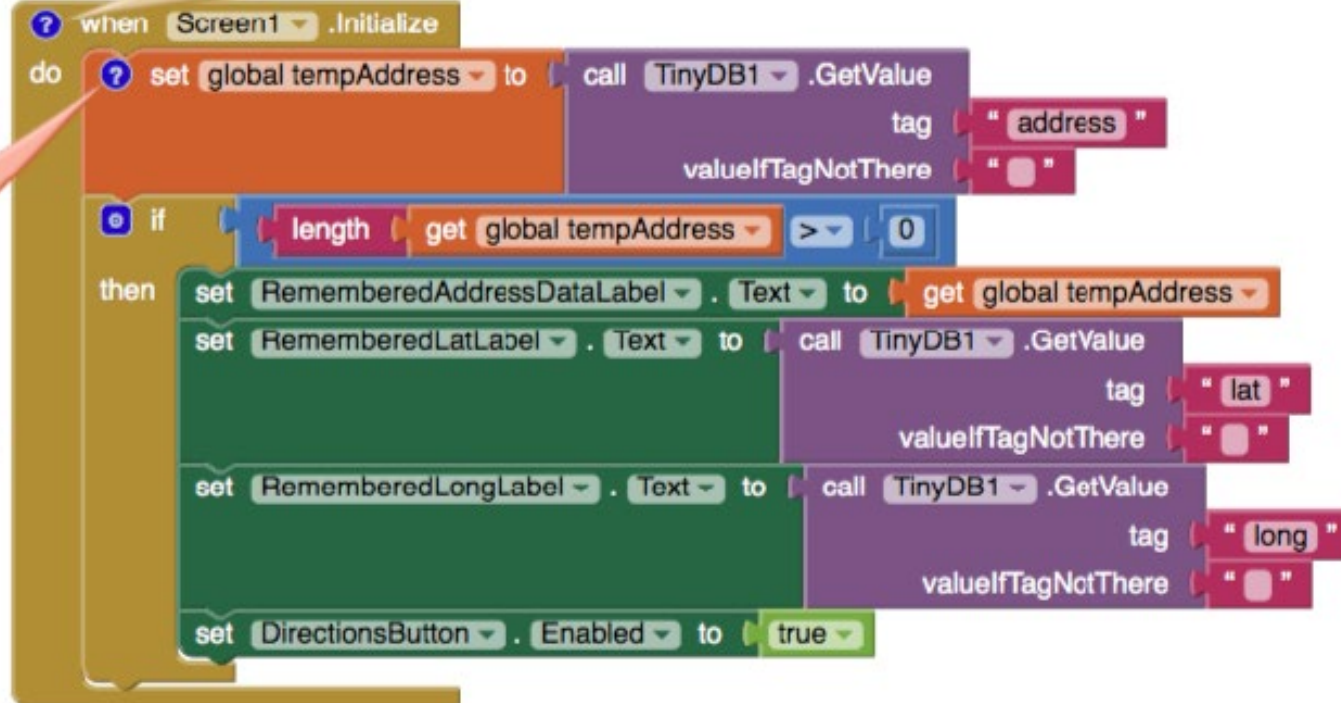
Once you have a reading, let the user "remember" it.

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.

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.

```
? when RememberButton .Click
do
  set RememberedAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set RememberedLatLabel . Text to LocationSensor1 . Latitude
  set RememberedLongLabel . Text to LocationSensor1 . Longitude
  call TinyDB1 .StoreValue
    tag "address"
    valueToStore LocationSensor1 . CurrentAddress
  call TinyDB1 .StoreValue
    tag "lat"
    valueToStore LocationSensor1 . Latitude
  call TinyDB1 .StoreValue
    tag "long"
    valueToStore LocationSensor1 . Longitude
  set DirectionsButton . Enabled to true
```

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`

The image shows a sequence of code blocks in a Scratch-style visual programming environment. The first block is a 'when DirectionsButton .Click' event. Inside a 'do' loop, there are three main blocks: a 'set ActivityStarter1 .DataUri to' block, a 'join' block, and a 'call ActivityStarter1 .StartActivity' block. The 'join' block is composed of several text blocks: a starting string 'http://maps.google.com/maps?saddr=', followed by 'CurrentLatLabel . Text', a comma separator, 'CurrentLongLabel . Text', an ampersand separator '&daddr=', 'RememberedLatLabel . Text', another comma separator, and 'RememberedLongLabel . Text'.

[Clip for APP Inventor 1](#)

פיתוח אפליקציות

שיעור מספר 3

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

- ▶ 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, a mobile emulator shows a screen with a text box containing the Hebrew word "טקסט" (Text) and a "Hello World" label below it. The top status bar shows the time as 9:48. In the center, the "Screen1" component tree lists "TextBox1" and "Button1". On the right, the "Properties" panel for "TextBox1" is visible, showing various settings such as BackgroundColor (Default), Enabled (checked), FontBold (unchecked), FontItalic (unchecked), FontSize (14.0), FontTypeface (default), Height (Automatic...), Width (Fill parent...), Hint (Hint for TextBox1), MultiLine (unchecked), NumbersOnly (unchecked), and Text (טקסט).

רמז - הטקסט הזה מופיע בתיבה כל עוד המשתמש לא התחיל להזין טקסט משלו

ניתן למנוע מהמשתמש להזין תווים - ספרות בלבד

ניתן לקלט טקסט במספר שורות

תוכן התחלתי של השדה - שימו לב לרב

LAYOUT - מיקום רכיבים על גבי המסך

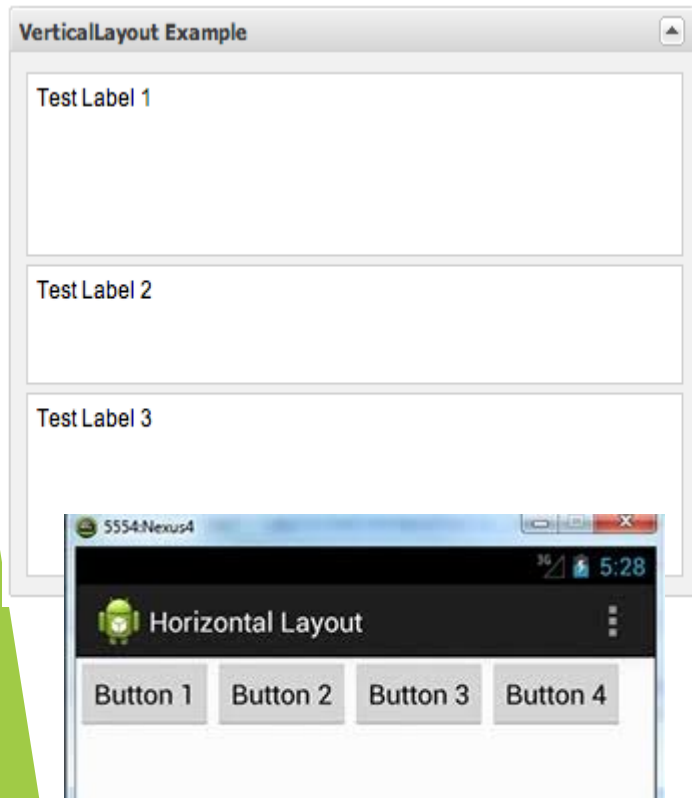
▶ על מנת לעזור לתוכניתן (לכם 😊) למקם את הרכיבים על גבי המסך קיימים רכיבים נוצרו הLAYOUTS.

▶ LAYOUTS - תפקידם לבצע סדר על גבי המסך.

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

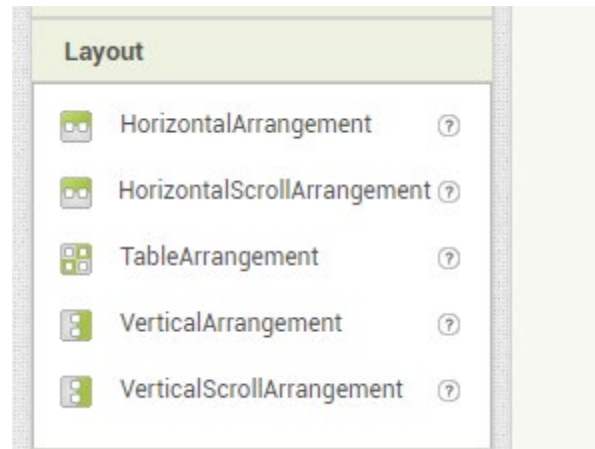
▶ בVERTICAL - כל רכיבים מסודרים אחד מתחת לשני

▶ 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.

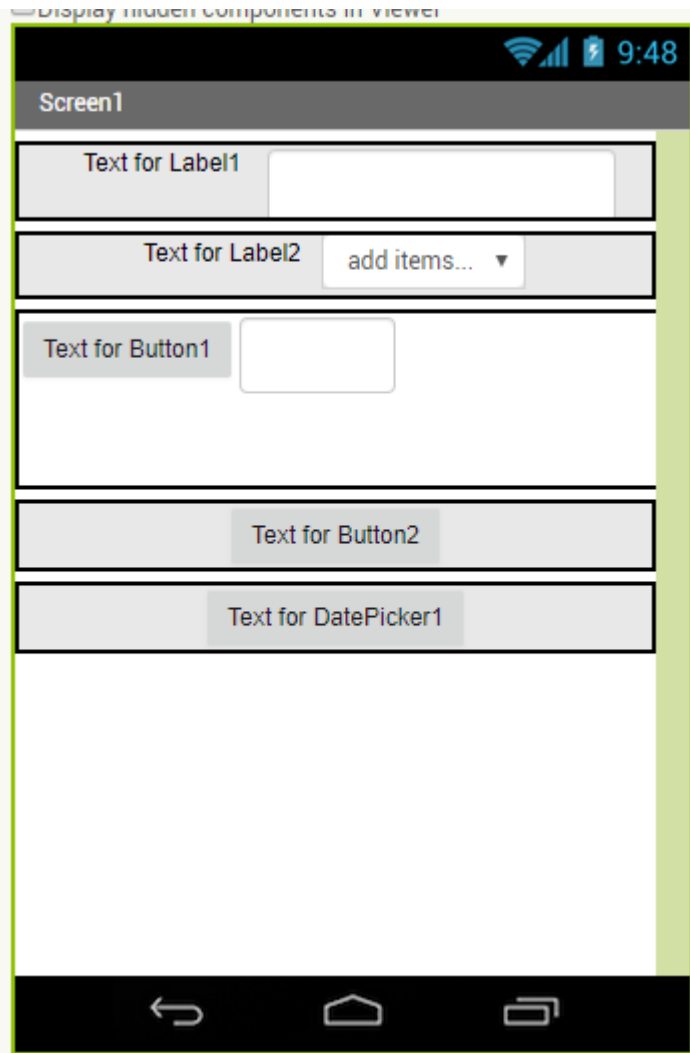
▶ אם המאפיין רוחב של `VerticalArrangement` מוגדר כ אוטומטי והוא ריק, הרוחב יהיה `100`.

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.

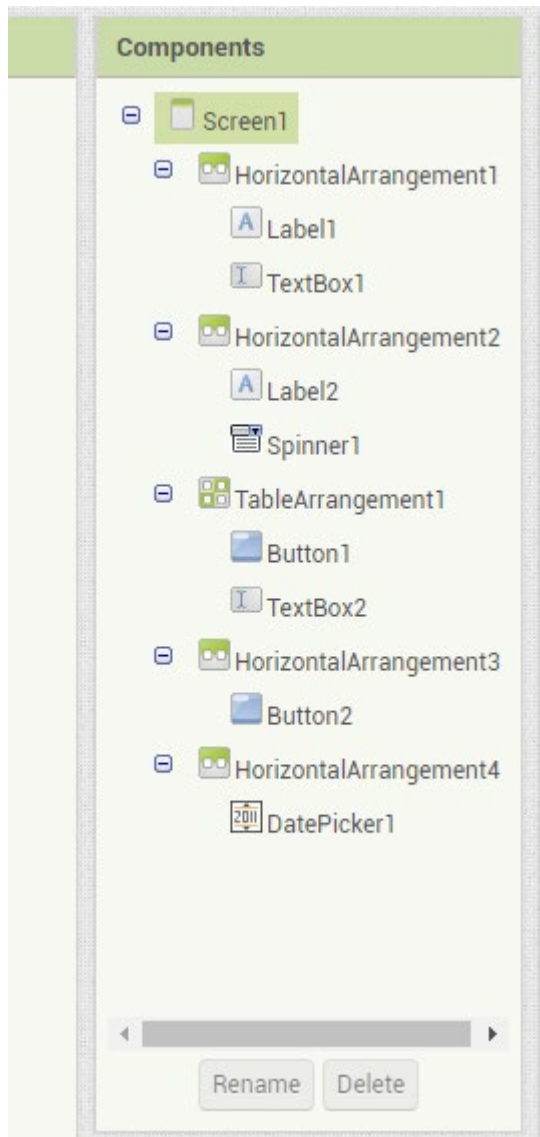
TABLE ARRANGMENT

- ▶ 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.



יש לבנות ולמקם רכיבים בצורה הבאה: ►

תרגיל מספר 4 - פתרון

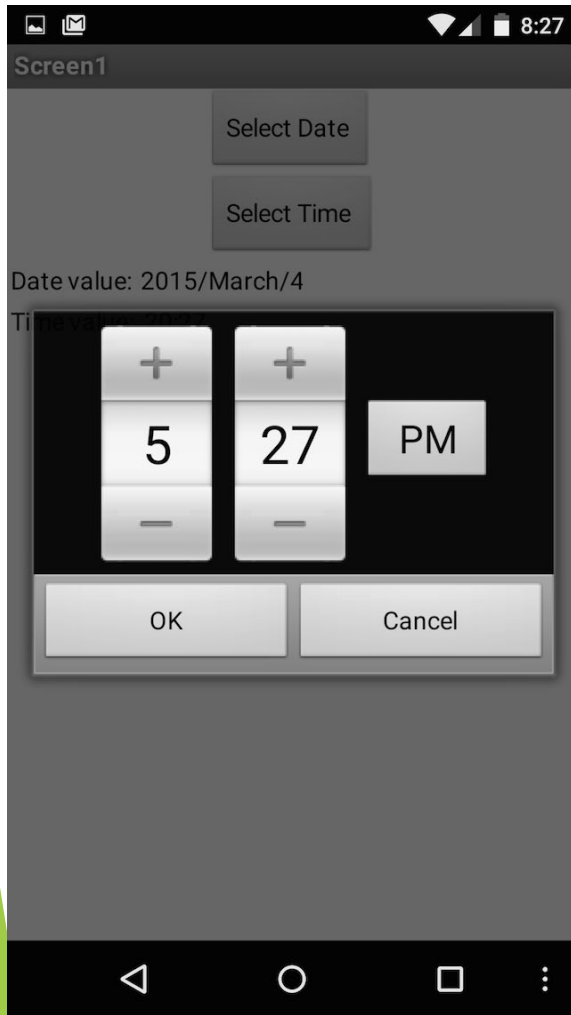


שימו לב - < על מנת למקם 2 רכיבים בשורה יצרנו 4 LAYOUT ים רוחביים.

בנוסף יצרנו LAYOUT אחד טבלאי.

בתוך LAYOUT ים הללו גררנו את הרכיבים שלנו.

תיבת בחירת תאריך DatePicker ותיבת בחירה זמן TIMEPICKER



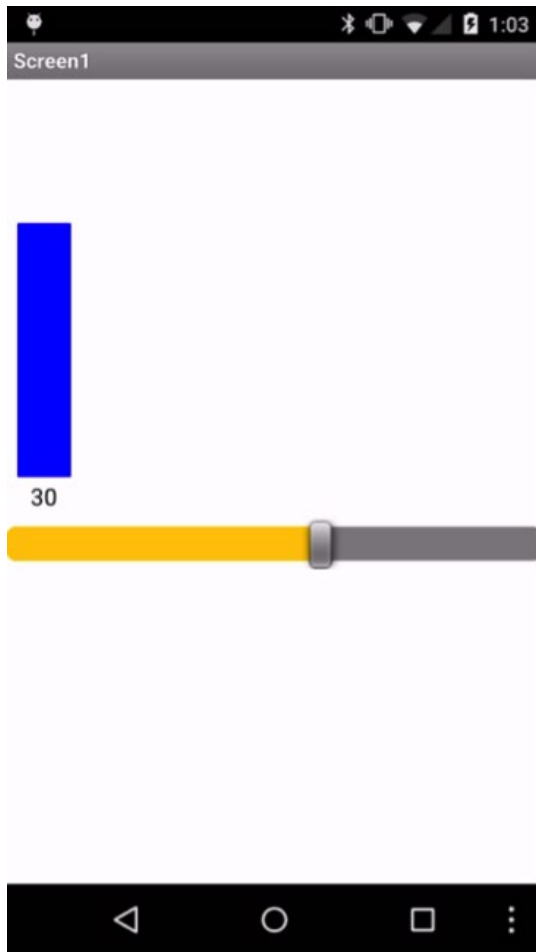
▶ לחצן שכאשר לחצת עליו, מציגה תיבת דו-שיח כדי לאפשר למשתמש לבחור את התאריך.

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

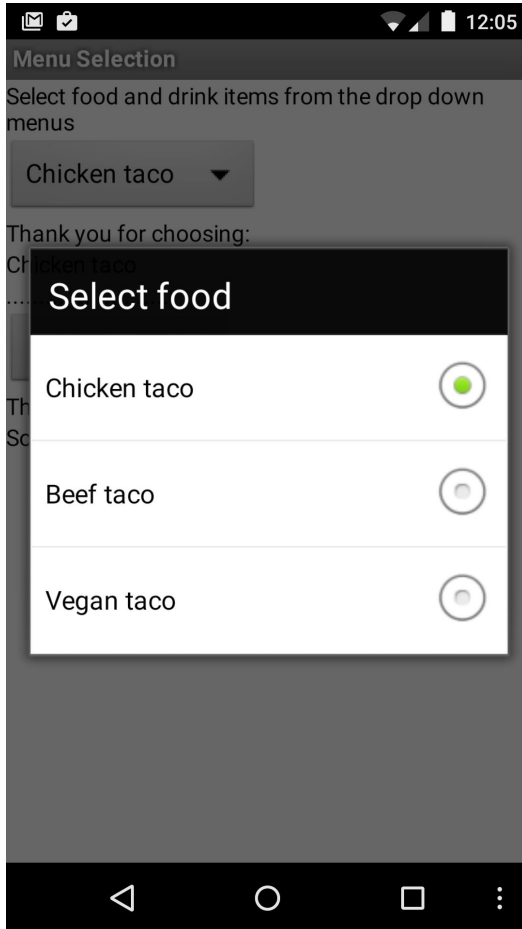
- ▶ בכל תזוזה של SLIDER, נורה אירוע POSITION CHANGED ומופיע בו מקום חדש.
- ▶ SLIDER מאפשר למשתמש דרך קלה לבחור ערך.
- ▶ SLIDER יש תכונות ערך מקסימלי וערך מינימלי וכמו כן יש ערך קפיצה בין מקסימלי למינימלי

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 1 - פרויקט מחשבון

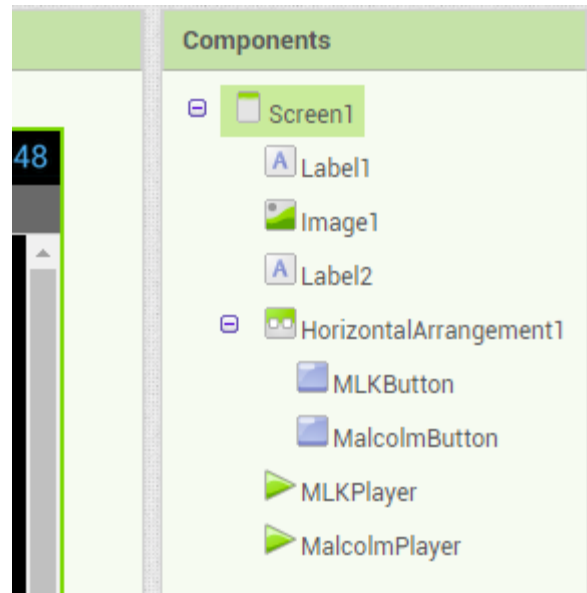
- ▶ אנחנו מתחילים לפתח פרויקט הראשון - מחשבון
- ▶ בשלב זה המחשבון צריך להראות ולא לחשב מספר בלבד - המחשבון יכיל כפתורים עבור כל ספרה (1,2,3,4,5,6,7,8,9,0)
- ▶ בנוסף יכיל המחשבון גם סימנים +, -, *, /
- ▶ וכמובן סימן =
- ▶ יש לסדר את הכפתורים בדומה למחשבון (כדאי להשתמש ב-LAYOUT)
- ▶ מעל הכפתורים יש להציג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

- ▶ בכל לחיצה על הכפתור ספרה, 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 IHaveADreamStarter - lesson 6

תרגיל חזרה קצר - הפתרון



פיתוח אפליקציות

שיעור מספר 4

שיעור 4 - משתנים ולוגיקה

.1 הרחבה בנושא BLOCK

.2 משתנה

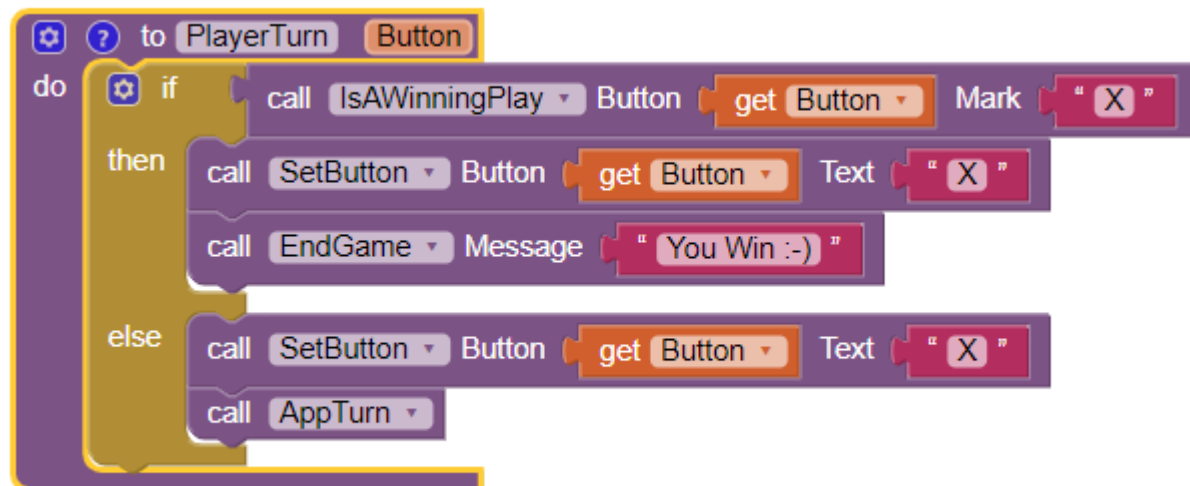
.3 תנאים

תרגיל חזרה קצר

- ▶ צור אפליקציה להזנת קוד סודי בכספת
- ▶ הלקוח צריך להציג קוד ובסיום #
- ▶ האפליקציה בכל לחיצה לא מציגה ללקוח את מספר שהוא הזין אלה 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...)



```
to PlayerTurn Button
do
  if call IsAWinningPlay Button get Button Mark "X"
  then
    call SetButton Button get Button Text "X"
    call EndGame Message "You Win :-)"
  else
    call SetButton Button get Button Text "X"
    call AppTurn
```

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

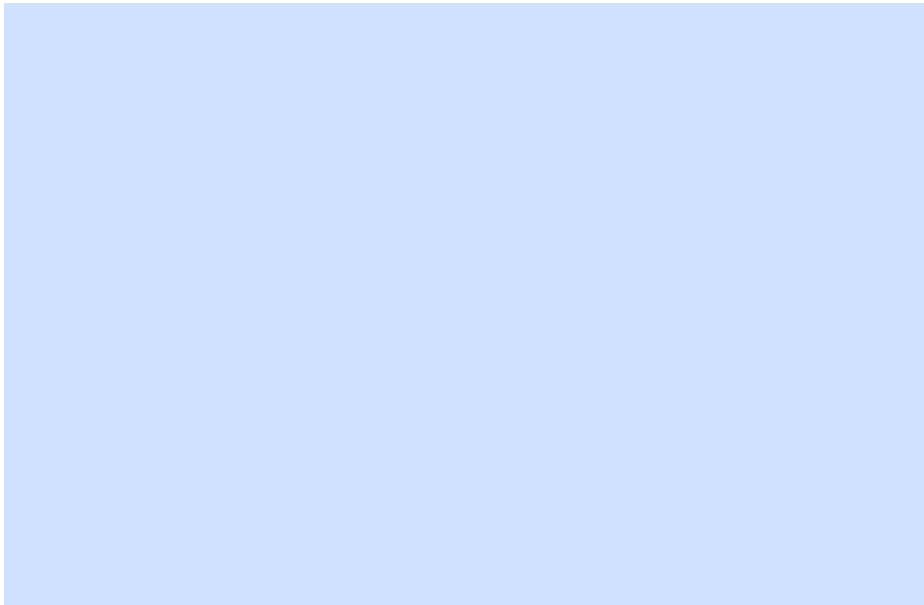
```
when NextButton.Click do  
  set Image1.Picture to
```

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

זכרון המחשב

כל תא בזכרון ממוספר
ושומר מספר אחד בלבד



כתובת

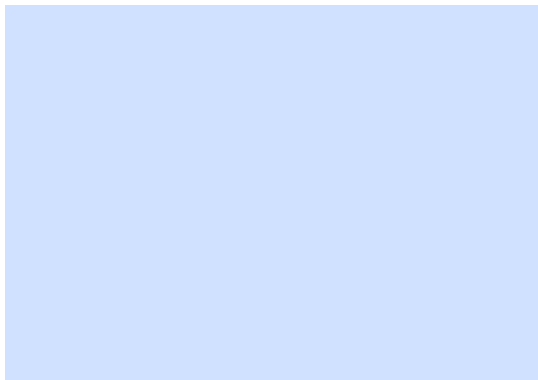
מספרים יכולים להשמר בגדלים שונים
מספרים קטנים BYTE צריכים תא אחד בזכרון

מספרים יותר גדולים - INTEGER
תופסים 2 תאים

מספרים הכי גדולים תופסים אפילו 4 תאים
ונקראים LONG ים - ארוכים

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.



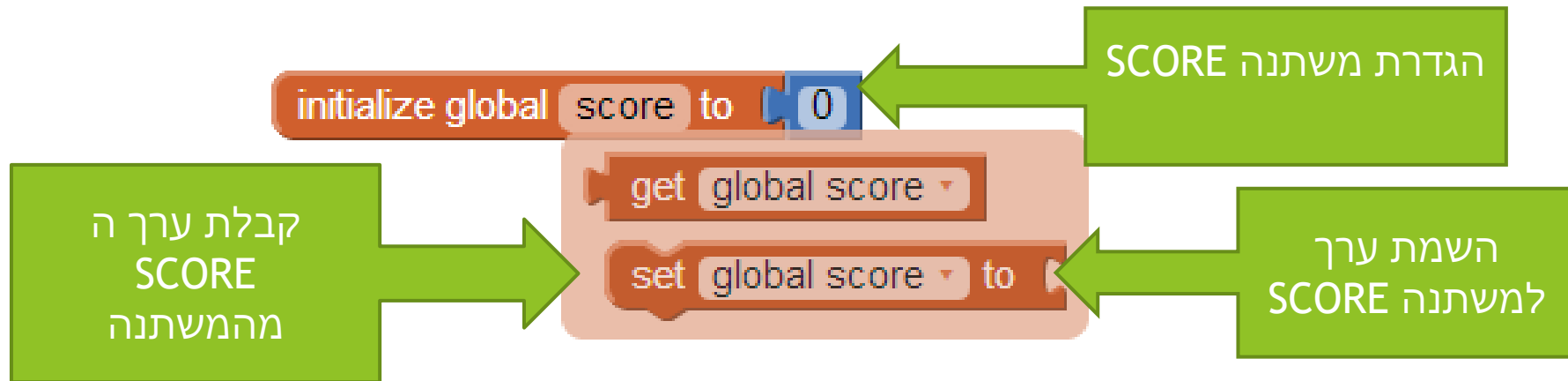
משתנים

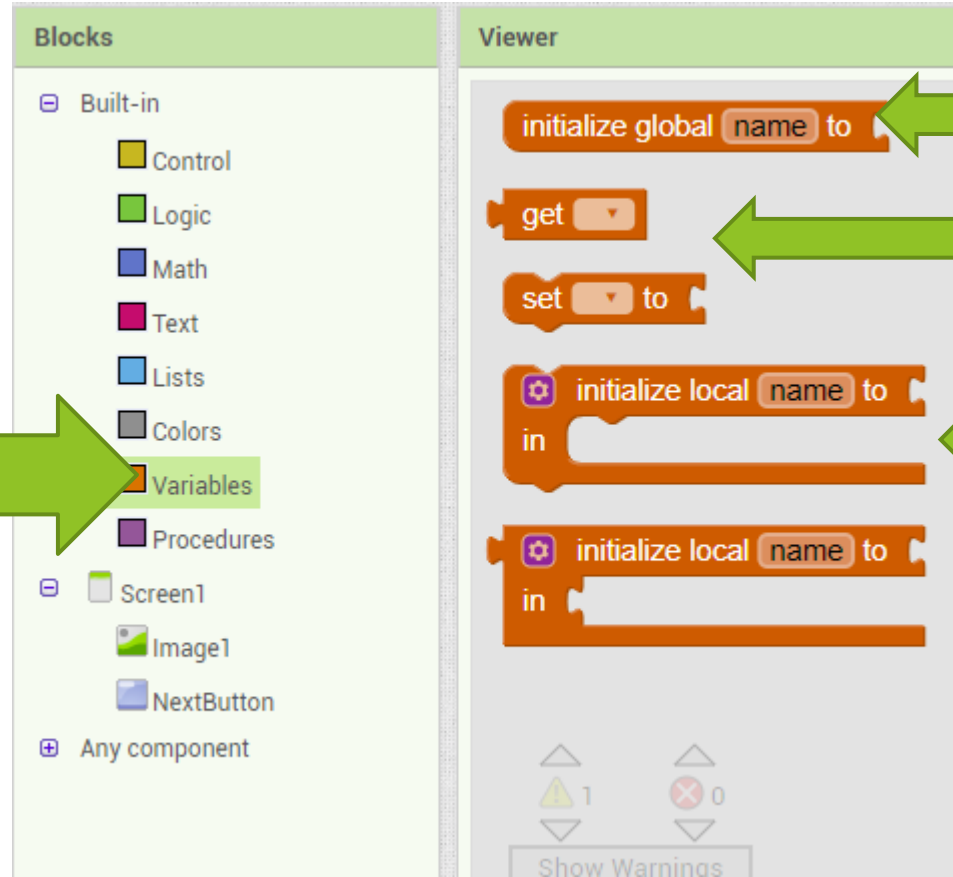
- ▶ כאשר אנחנו מדברים על המשתנים ב APP INVERTOR , אנחנו מתכוונים לכל הגורמים היודעים לנהל ערכים
- ▶ בפקדים ורכיבים - רכיבי TEXT
- ▶ ניתן גם להגדיר משתניים גלובליים ומקומיים
- ▶ כל עבודה עם המשתנים ב APP INVERTOR מתבצעת בחלק BLOCK

משתנים

- ▶ ב-APP INVERTOR המשתנים יכולים להישמר ברמה גלובאלית
- ▶ ז"א הם נשמרים בבלוק הגלובאלי - < שנקרא GLOBAL
- ▶ ברגע שמתנה 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 ראשונית - score=0

משתנים





בחירת VARIABLES

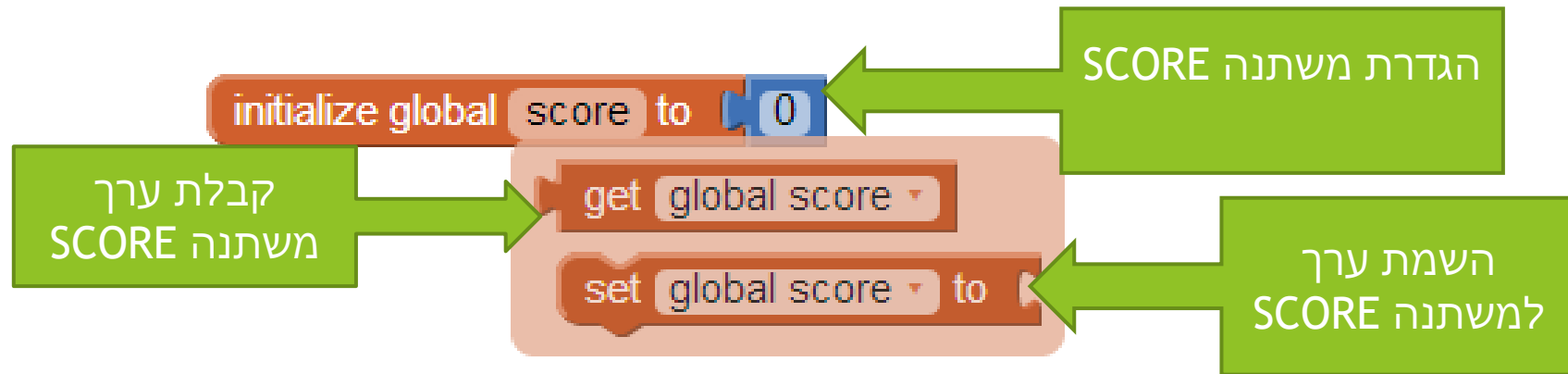
הגדרת משתנה

קבלת ערך מהמשתנה

BLOCK שבו מוגדר משתנה NAME בלוק מופעל בלולאה

הגדרת משתנה

משתנים



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 1 - 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, אלא גם משתנה אחר למשל אם i ו- j הם מספריים
- ▶ למשל אם קבענו ש $i=1$ אז ניתן לכתוב $i=j$ 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 'Blocks' and 'Viewer' panes. The 'Blocks' pane on the left shows the 'Math' category selected. The 'Viewer' pane on the right shows a script with several math blocks. Green callout boxes with arrows point to specific blocks, each with a Hebrew label. Three inset boxes on the right show the state of a variable 'x1' after different operations.

Annotations:

- השמת מספר כלשהו למשתנה (Assigning any number to a variable)
- חיבור בין 2 משתנים (Addition of 2 variables)
- חיסור בין שני המשתנים (Subtraction of two variables)
- כפל בין שני המשתנים (Multiplication of two variables)
- חילוק בין 2 המשתנים (Division of two variables)
- חזקה (Power)

Inset Boxes:

- Initial state: initialize local x1 to 222
- After addition: set x1 to 333 + 222
- After subtraction: set x1 to 333 - 111

משתנים

דגשים לעבודה עם MATH

- בכל שימוש במספר יש להשתמש ב []
- בכל פעולת השמה נדרש להשתמש ב SET
- 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 $X=X+222$

```
when Screen1.Initialize
do
  initialize local x1 to 222
  in set x1 to [get x1 + 222]
```

תרגיל מספר 1 - משתנים

- ▶ כתוב אפליקציה לצורך ספירת כניסות למזיאון. 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

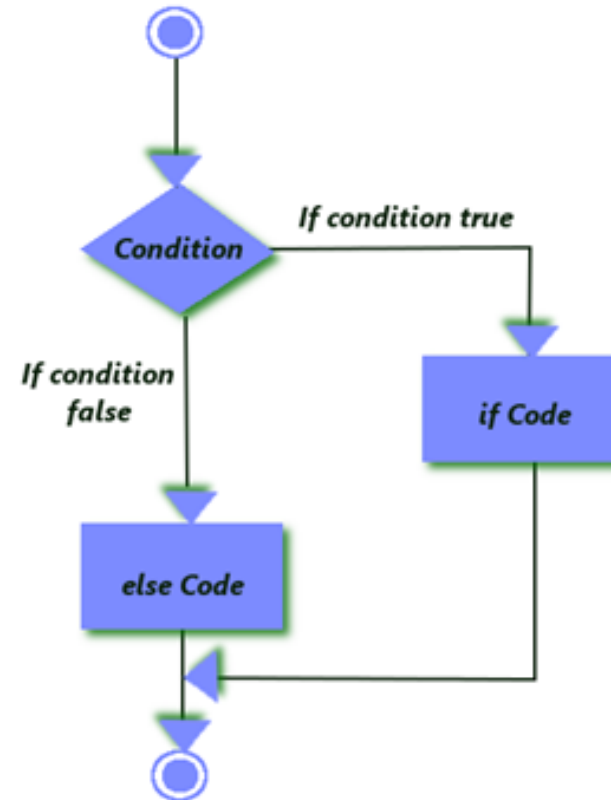
תנאים

- ▶ 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

Variables

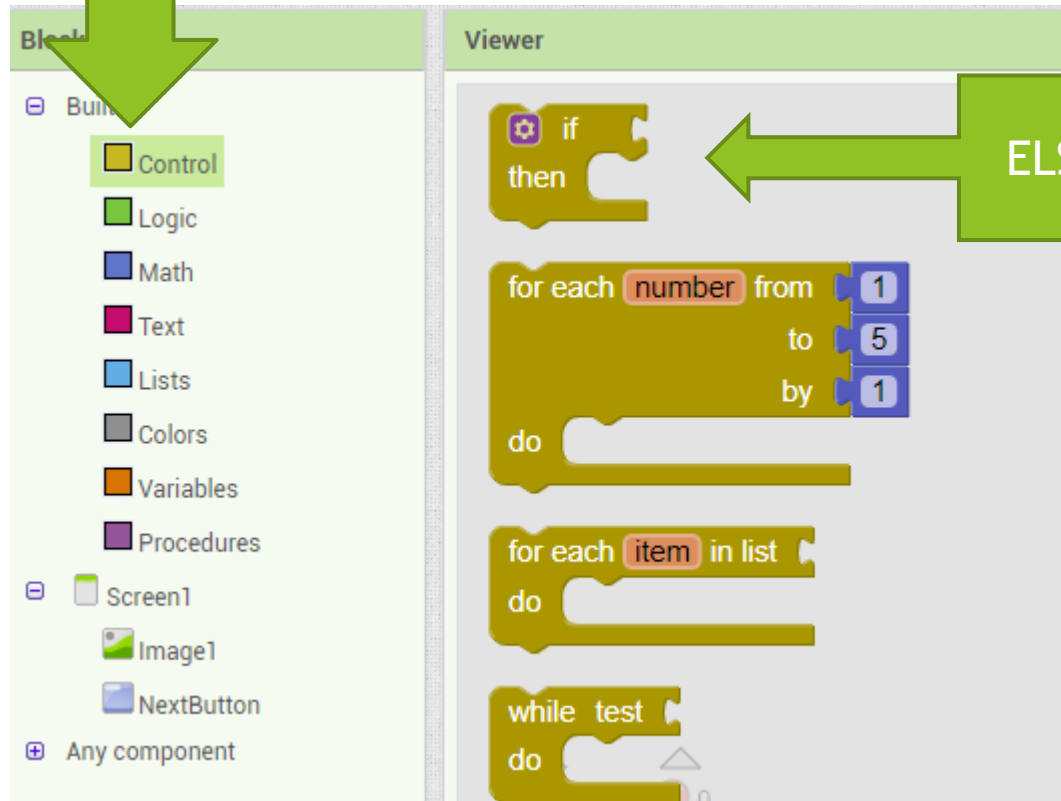
תנאים

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 -

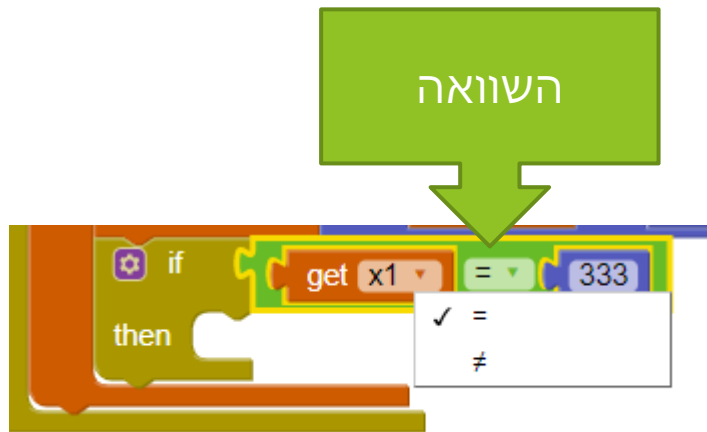


גוררים את IF וELSE

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 ל 1 אז מעדכנים את התמונה לשם א
- ▶ אם ערך שווה ל 2 אז מעדכנים את התמונה לשם ב וכך הלאה
- ▶ נדבר על התמונות במצגות הבאות

```
if [get global Counter] = 1
  then set Button1 Image to "duckhat1.png"
else if [get global Counter] = 2
  then set Button1 Image to "duckhat2.png"
else if [get global Counter] = 3
  then set Button1 Image to "duckhat3.png"
  set global Counter to 0
else set Button1 Image to "duckhat4.png"
```


תנאים

The image shows the Scratch 'Blocks' and 'Viewer' panels. The 'Blocks' panel on the left contains a category menu with 'Text' and 'Lists' visible. The 'Viewer' panel on the right displays a stack of logical blocks: 'true', 'false', 'not', '=', 'and', and 'or'. Green callout boxes with arrows point to these blocks, providing Hebrew descriptions:

- Block: 'true' → Annotation: 'ערך TRUE - אמת' (TRUE value - true)
- Block: 'false' → Annotation: 'ערך FALSE שקר' (FALSE value - false)
- Block: 'not' → Annotation: 'השוואה בין ערכים' (Comparison between values)
- Block: '=' → Annotation: 'השוואה בין ערכים' (Comparison between values)
- Block: 'and' → Annotation: 'השוואה בין ערכים' (Comparison between values)
- Block: 'or' → Annotation: 'השוואה בין ערכים' (Comparison between values)

Additional annotations on the left side of the 'Blocks' panel:

- Annotation: 'ערך FALSE שקר' (FALSE value - false) with an arrow pointing to the 'true' block.
- Annotation: 'מספר תנאים בלולאה מחוברים' (Number of conditions in a loop connected) with an arrow pointing to the 'and' and 'or' blocks.

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 גלובאלי FIRST
- כאשר המשתמש לוחץ על סימן יש לשמור אות במשתנה גלובאל SIGN
- כאשר המשתמש לוחץ על מספר השני - יש לשמור אותו במשתנה גלובאלי 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 ■ - 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

The image shows a screenshot of the Scratch interface, specifically the Blocks palette and the Script area. The Blocks palette is on the left, and the Script area is on the right. Several text boxes with arrows point to specific blocks, explaining their functions in Hebrew.

Annotations:

- חיבור בין מחרוזות -** (Joining strings) - points to the "join" block.
- ניתן לכתוב מחרוזת** (You can write a string) - points to the "say" block.
- קבלת גודל מחרוזת** (Getting string length) - points to the "length" block.
- השוואה איזה מחרוזת גדולה יותר** (Comparing which string is longer) - points to the "compare texts" block.
- הפיכת כל האותיות לאותיות הגדולות - לא רלוונטי אם עובדים בעברית** (Converting all letters to uppercase - irrelevant if working in Hebrew) - points to the "uppercase" block.
- בחירה ברכיב טקסט בבלוק** (Choosing a text component in the block) - points to the "Text" category in the Blocks palette.

Blocks Palette:

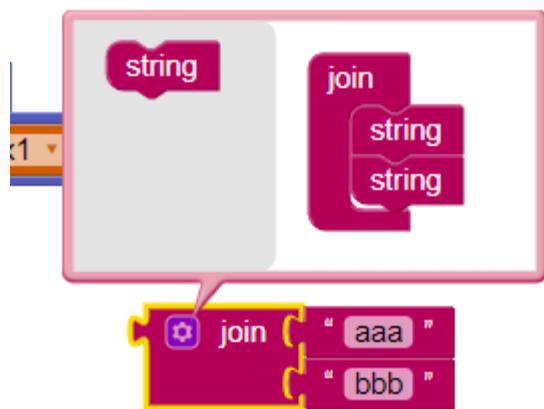
- Math
- Text** (highlighted)
- Lists
- Colors
- Variables
- Procedures
- Screen1
- Image1
- NextButton
- Any component

Script Area:

- say
- join
- length
- is empty
- compare texts
- trim
- uppercase
- starts at text piece
- contains text piece

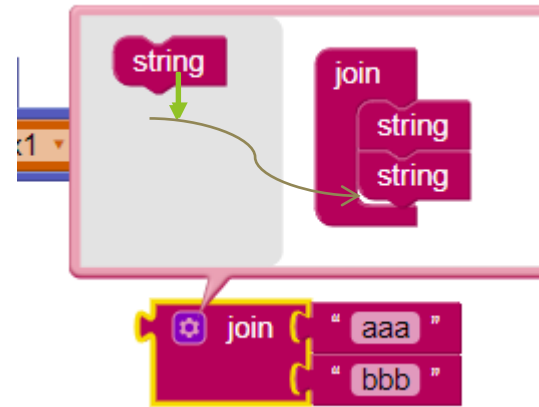
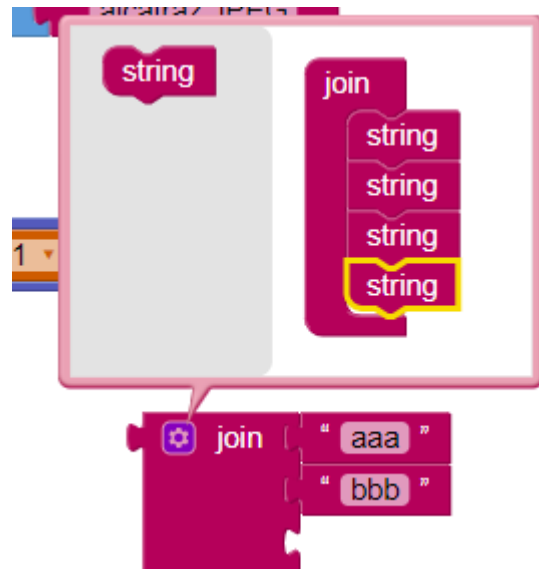
משתנה טכסט - JOIN

- ▶ 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!!!

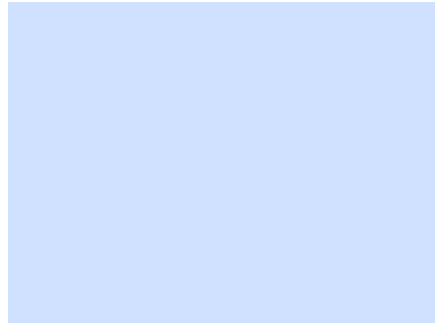


תרגיל מספר 1

- 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

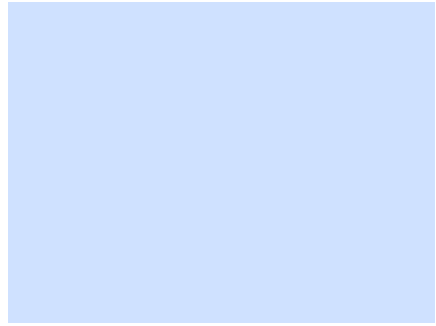
משתנה טכסט - JOIN - הרחבת פעולה

לאחר הגרירה ניתן לבנות ממש מחרוזות ארוכים ▶



משתנה טכסט - JOIN - הרחבת פעולה

לאחר הגרירה ניתן לבנות ממש מחרוזות ארוכות



- ▶ רשימה 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 screenshot shows a programming environment with a 'Built-in' category on the left. A green box labeled 'בניית רשימה קיימת' (Building an existing list) has an arrow pointing to the 'make a list' block. Another green box labeled 'יצירת רשימה ריקה' (Creating an empty list) has an arrow pointing to the 'create empty list' block. A third green box labeled 'הוספת ערכים לרשימה קיימת' (Adding items to an existing list) has an arrow pointing to the 'add items to list' block. The 'Lists' category in the sidebar includes blocks for: 'create empty list', 'make a list', 'add items to list', 'is in list?', 'length of list', 'is list empty?', 'pick a random item', and 'index in list'. The 'add items to list' block has two input fields labeled 'list' and 'item'.

רשימה - LISTS

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

יצירת
משתנה
גלובאלי



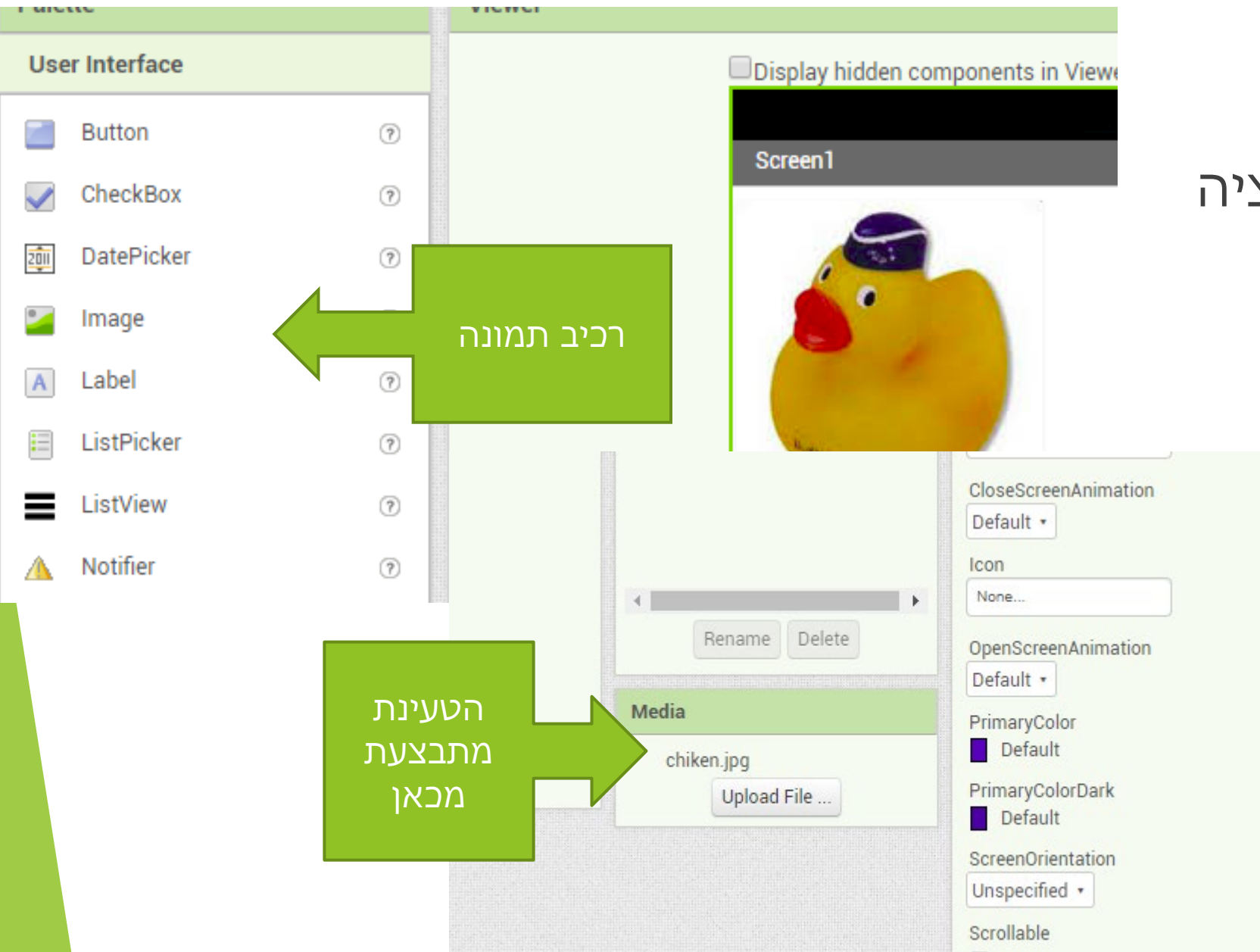
בניית
רשימה של
שמות

יצירת משתנה
בשם
IMAGEFILES

תמונה

- ▶ 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



רכיב תמונה

הטעינת
מתבצעת
מכאן

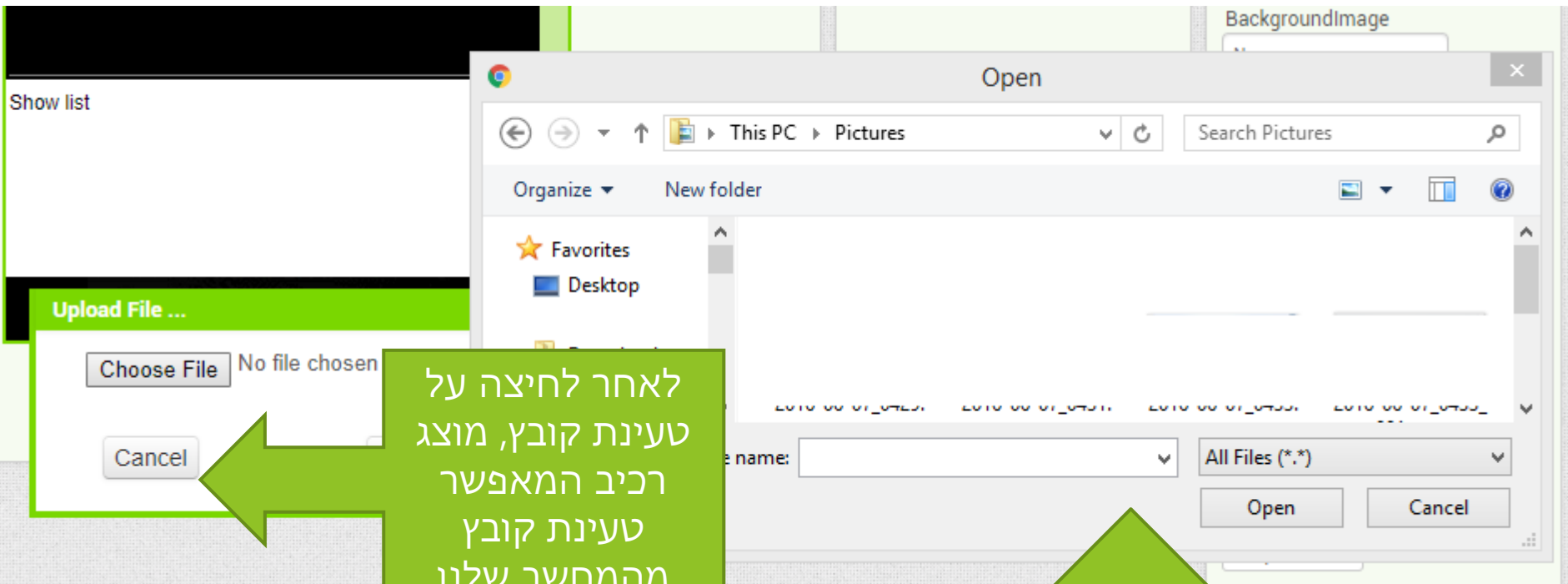
לפני שמציגים את
התמונה

יש לטעון אותה לאפליקציה

לא חייבים להשתמש
בתמונות שטענו

(אבל כמובן לא כדאי
להחזיק באפליקציה
תמונות מיותרות)

טעינת התמונה



לאחר לחיצה על
טעינת קובץ, מוצג
רכיב המאפשר
טעינת קובץ
מהמחשב שלנו

יש לבחור קובץ שהכנו
מראש!!!

רכיב IMAGE

The screenshot shows the IDE interface. On the left, the 'Components' panel lists 'Screen1', 'Image1', and 'NextButton'. On the right, the 'Properties' panel for 'Image1' includes fields for 'Height' (Automatic...), 'Width' (Automatic...), 'Picture' (None...), 'RotationAngle' (0.0), 'ScalePictureToFit' (checkbox), and 'Visible' (checkbox checked).

התאמה לגודל -
לאחר בחירת תמונה,
מאוד נוח ייקטין את
התמונה לגודל
המסך

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

The screenshot shows the 'Media' panel with a list of image files: 'alcatraz.JPEG', 'coitTower.jpg', and 'ggBridge.jpg'. There are 'Rename' and 'Delete' buttons above the list, and an 'Upload File ...' button below it.

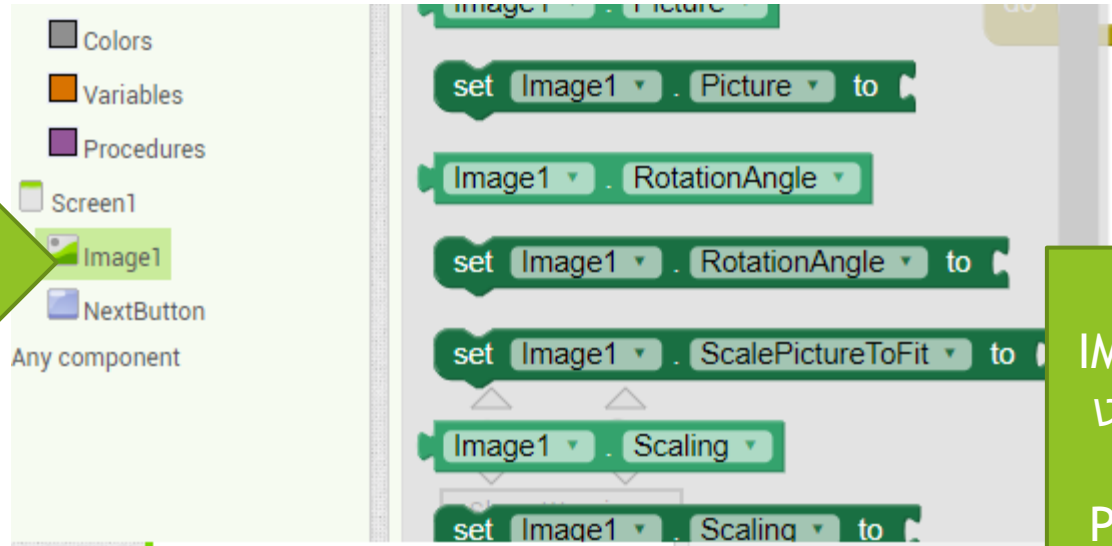
ניתן לבחור רק את
התמונות הללו להיות
מוצגים

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...

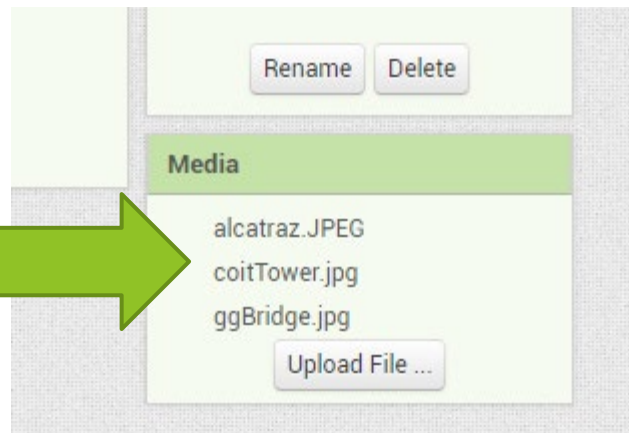
שימוש ועדכון IMAGE בחלק של הBLOCKS

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



Show the next picture when button clicked



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 3 - הצגת תמונות

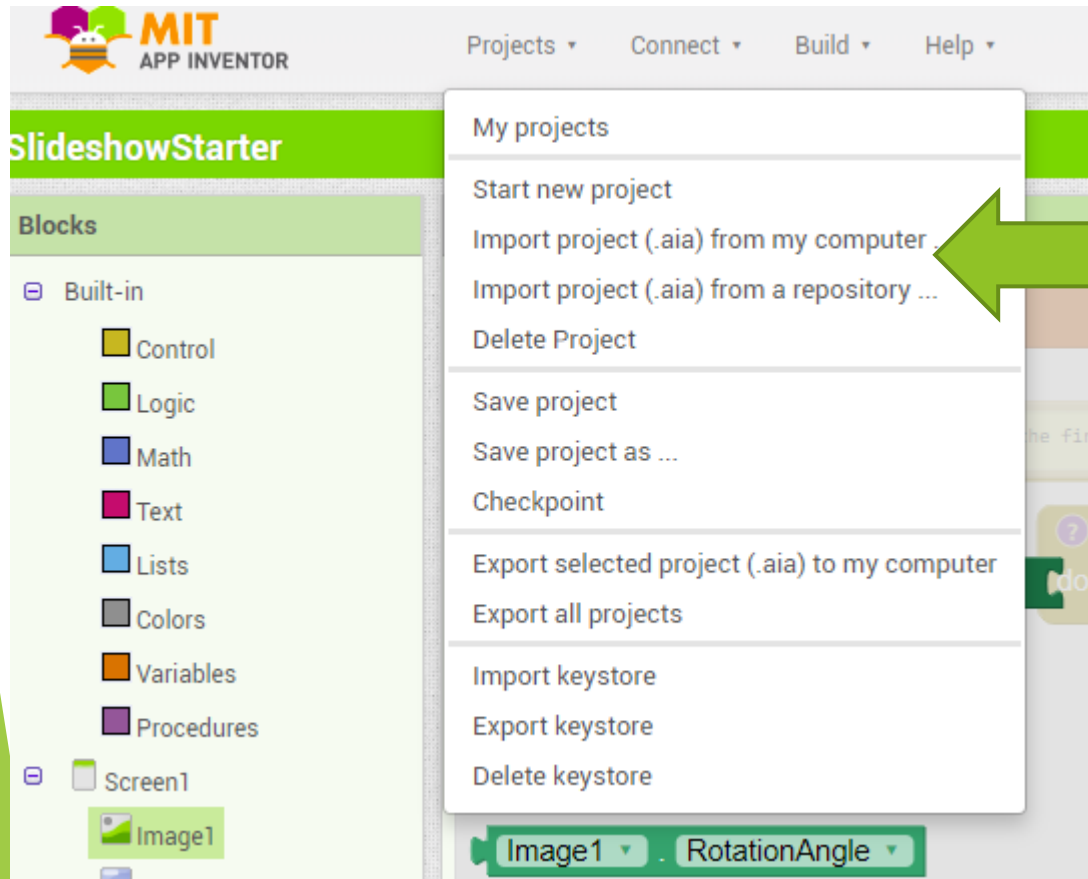
- יש ליצור אפליקציה המציגה 3 תמונות (אשר הכנתם מראש)
- יש להכין מסך עם התמונה (כאשר מראש מוצגת תמונה ראשונה מתוך 3 - ניתן להציגה מעל רכיב IMAGE או כפתור)
- יש להוסיף למסך זה 3 כפתורים
- בלחיצה על כל כפתור, תתחלף התמונה המוצגת ותוצגנה תמונה הבאה.

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 ל APP INVERTOR אפליקציה קיימת, יש להכין את הקובץ במחשב (להוריד אותו)
- ▶ למשל הכנו לכם קובץ של אפליקציה מסוימת...



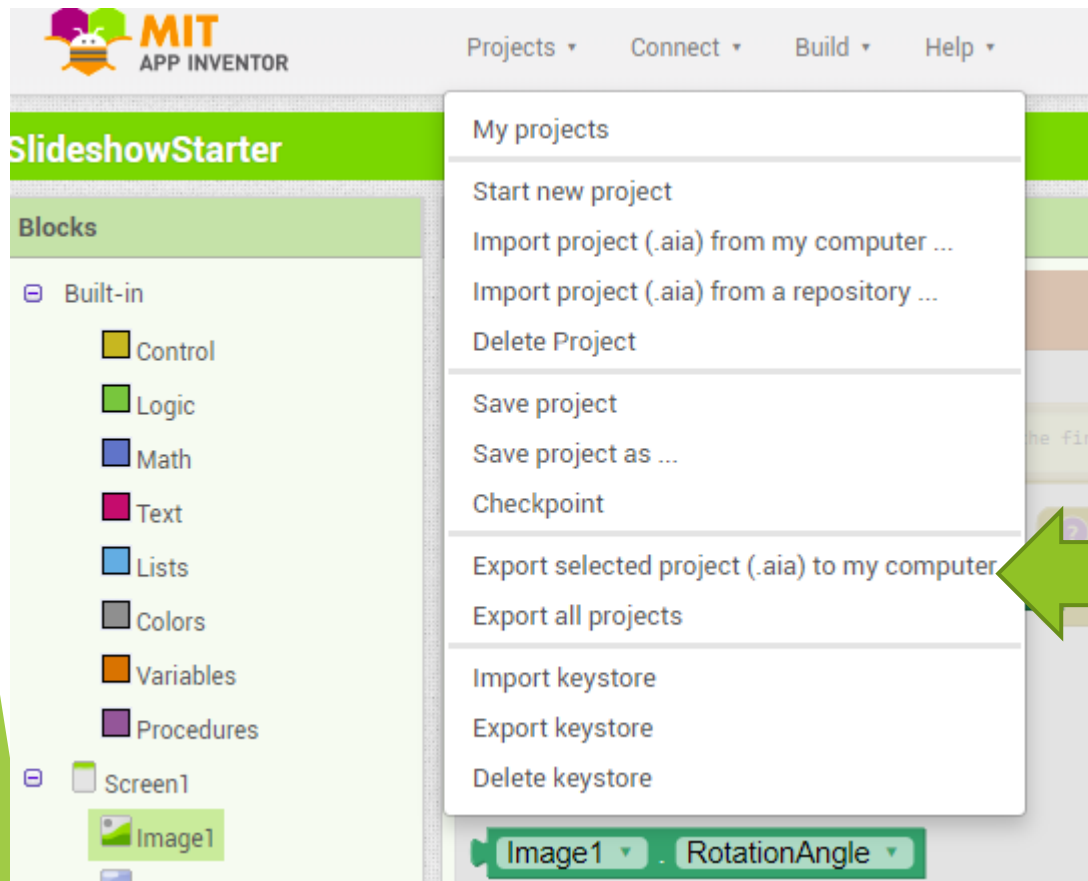
יש לבחור באופציה
טעינת הקובץ מהמחשב

קבצים של APP INVERTOR נשמרים עם סיומת

AIA

שמירת אפליקציה למחשב

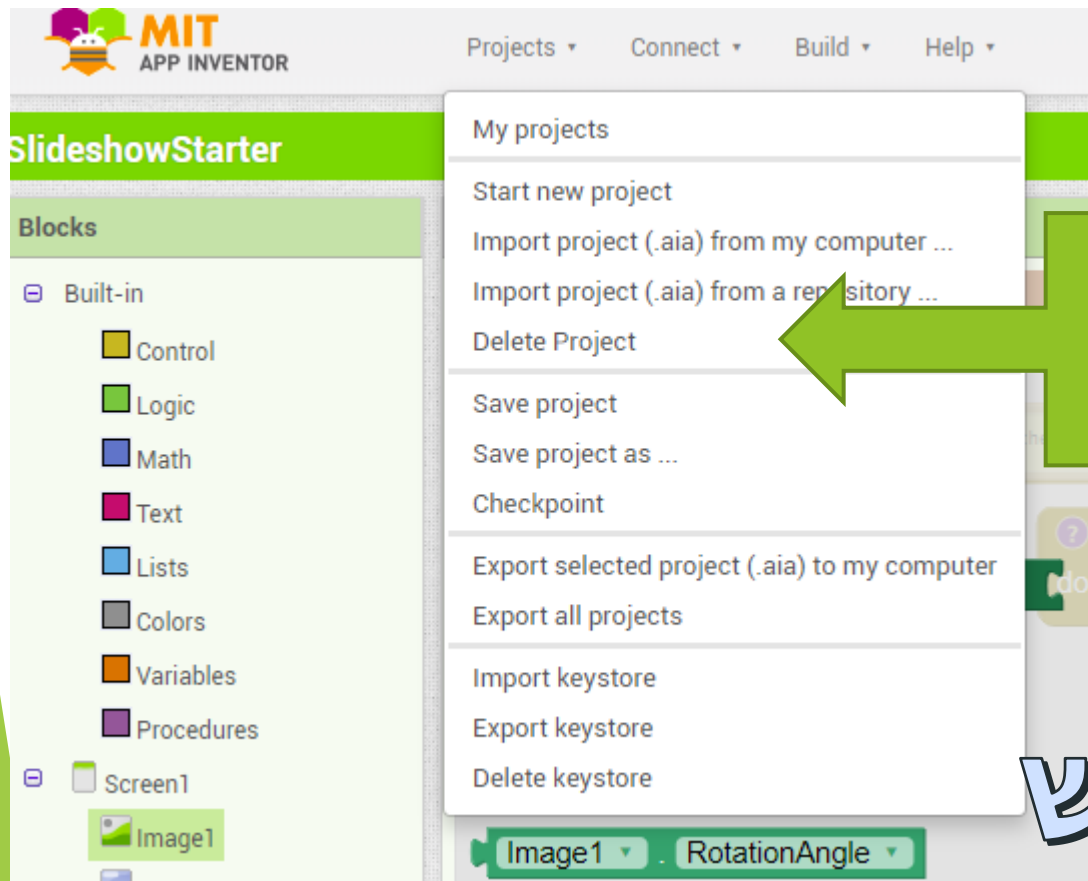
- ▶ ניתן גם לשמור את האפליקציה שפיתחנו חזרה למחשב... תחת קובץ AIA
- ▶ ניתן גם לשמור את האפליקציות ברשימת האפליקציות ב APP INVERTOR
- ▶ האפליקציות נשמרות תחת משתמש
- ▶ GMAIL שלנו



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 1 - שמירה וטעינת אפליקציות

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

- ▶ 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 1 - מצגת תמונות

Viewer

initialize global imageFiles to

make a list

- coitTower.jpg
- ggBridge.jpg
- alcatraz.JPEG

הרשימה שהכנו

Show the first picture when app begins

when Screen1.Initialize

do

טעינה ראשונית לרכיב תמונה

Show the next picture when button clicked

when NextButton.Click

do

set Image1.Picture

טעינת המשך לתמונה

Show Warnings 1 0

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

פיתוח אפליקציות

שיעור מספר 6

שיעור ■ - לולאה והמשך תמונות

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

תרגיל ■ - תרגיל חזרה קטן

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



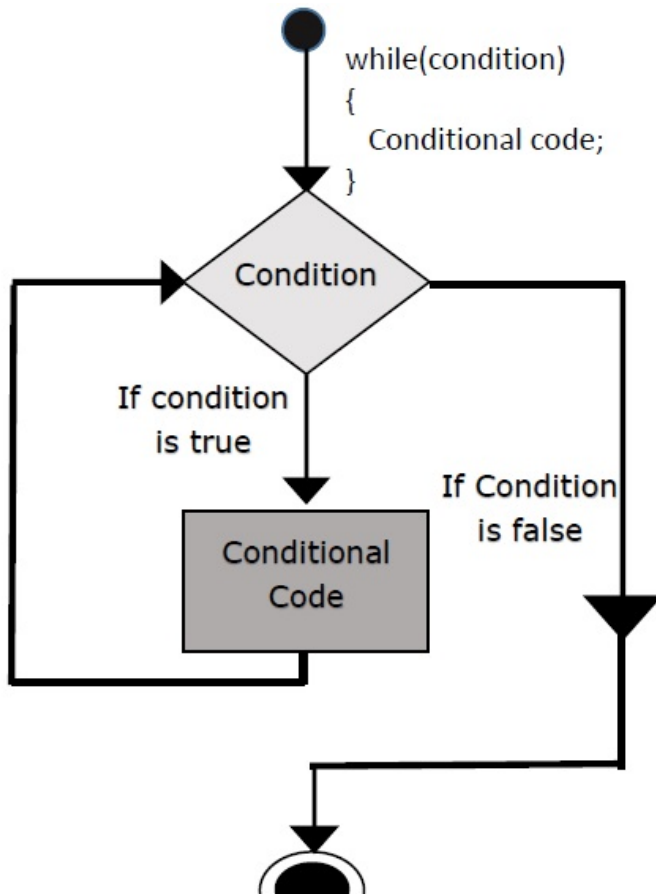
▶ היכולת החזקה הידוע של המחשב היא לבצע פעולות חוזרות

▶ 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



לולאות

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

The screenshot shows a programming environment with two main panels: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' panel is divided into 'Built-in' and 'Screen1' categories. Under 'Built-in', the 'Control' category is highlighted in green. Other categories include Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under 'Screen1', there are components for Image1 and NextButton. The 'Viewer' panel displays a sequence of code blocks: a 'for each number from 1 to 5 by 1' loop, a 'for each item in list' loop, a 'while test' loop, an 'if then else' block, and a 'do' block. A green arrow points from the 'Control' category in the 'Blocks' panel to the 'while test' block in the 'Viewer' panel.

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



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



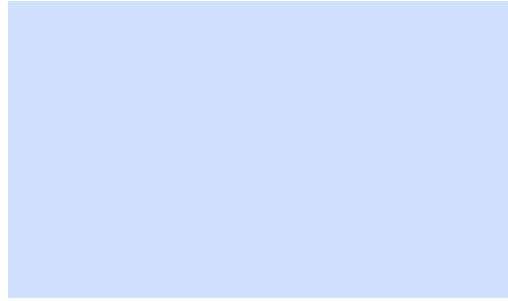
- ▶ לולאת FOR - לולאת FOR דומה ללולאת WHILE. ההבדל הוא שבמקרה של FOR אין צורך לקדם את המשתנה התנאי
- ▶ אם אנו רוצים לעשות 5 צעדים אז מגדירים את number ל 1
- ▶ לאחר מכן מקדמים את number כל פעם ב 1 עד שהוא מגיע להיות 5
- ▶ כל הפעולות המופיעות ב DO, מתבצעות 5 פעמים

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 הרבה יותר מהר

לולאות FOR

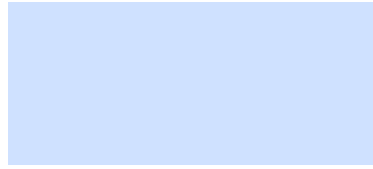


- ▶ ניתן לשלב TIMER בעבודה עם לולאת FOR
- ▶ אחרי שילוב TIMER יהיה עיקוב של כמה שניות במעבר משלב לשלב
- ▶ ביצוע שלב אחד בלולאת FOR נקרא אינטרציה INTERATION



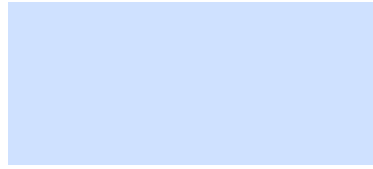
- ▶ כתוב אפליקציה אשר שקולטת שני מספרים . האפליקציה מציגה כמות * לפי מספרים שהוגדרו
- ▶ למשל אם נקלטו מספרים 1 ו5 , אפליקציה תדפיס 5 כוכביות
- ▶ למשל אם נקלטו מספרים 1 ו10, תציג 10 כוכביות

לולאות FOREACH



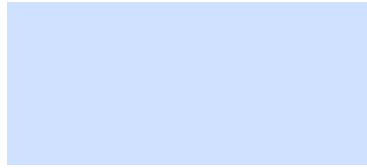
- ▶ לולאת FOREACH - לולאת FOREACH דומה קצת ללולאת FOR בכך שגם היא מתקדם את כמות הצעדים בצורה אוטומטית
- ▶ לולאת FOREACH - עוברת על כל LIST ומבצעת עבור כל איבר את פעילות

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

```
when PreviousButton .Click
do
  if [get global index > 1]
  then
    set global index to [get global index - 1]
    set Image1 . Picture to [select list item list index get global imageFiles get global index]
```

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 שיעון 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"
```

Properties

Clock1

TimerAlwaysFires

TimerEnabled

TimerInterval 1000

הפעלת שעון מראש

הגדרת קפיצה שעון כל שנייה

- ▶ יש לפתח אפליקציה המציגה 3 TEXTBOX
- ▶ כל TEXTBOX מופיע (הופל לVISIBLE) לאחר 3 שניות אחרי הקודם
- ▶ תרגיל די פשוט ולכן מי שמסיים מוזמן לתרגיל מסכם של היום - 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 1 - מצגת תמונות

Viewer

```
initialize global imageFiles to  
  make a list  
    "coitTower.jpg"  
    "ggBridge.jpg"  
    "alcatraz.JPEG"  
  
Show the first picture when app begins  
when Screen1.Initialize do  
  
Show the next picture when button clicked  
when NextButton.Click do  
  set Image1.Picture
```

הרשימה שהכנו

טעינה ראשונית לרכיב תמונה

טעינת המשך לתמונה

Show Warnings 1 0

- ▶ 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

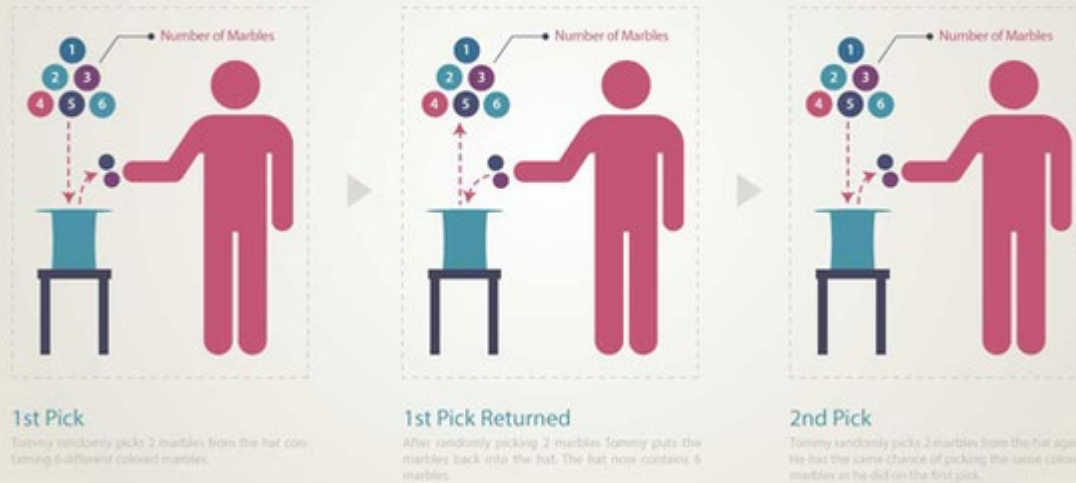
שיעור ■ - מספר אקראי, סנסורים וטריוויה

- .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

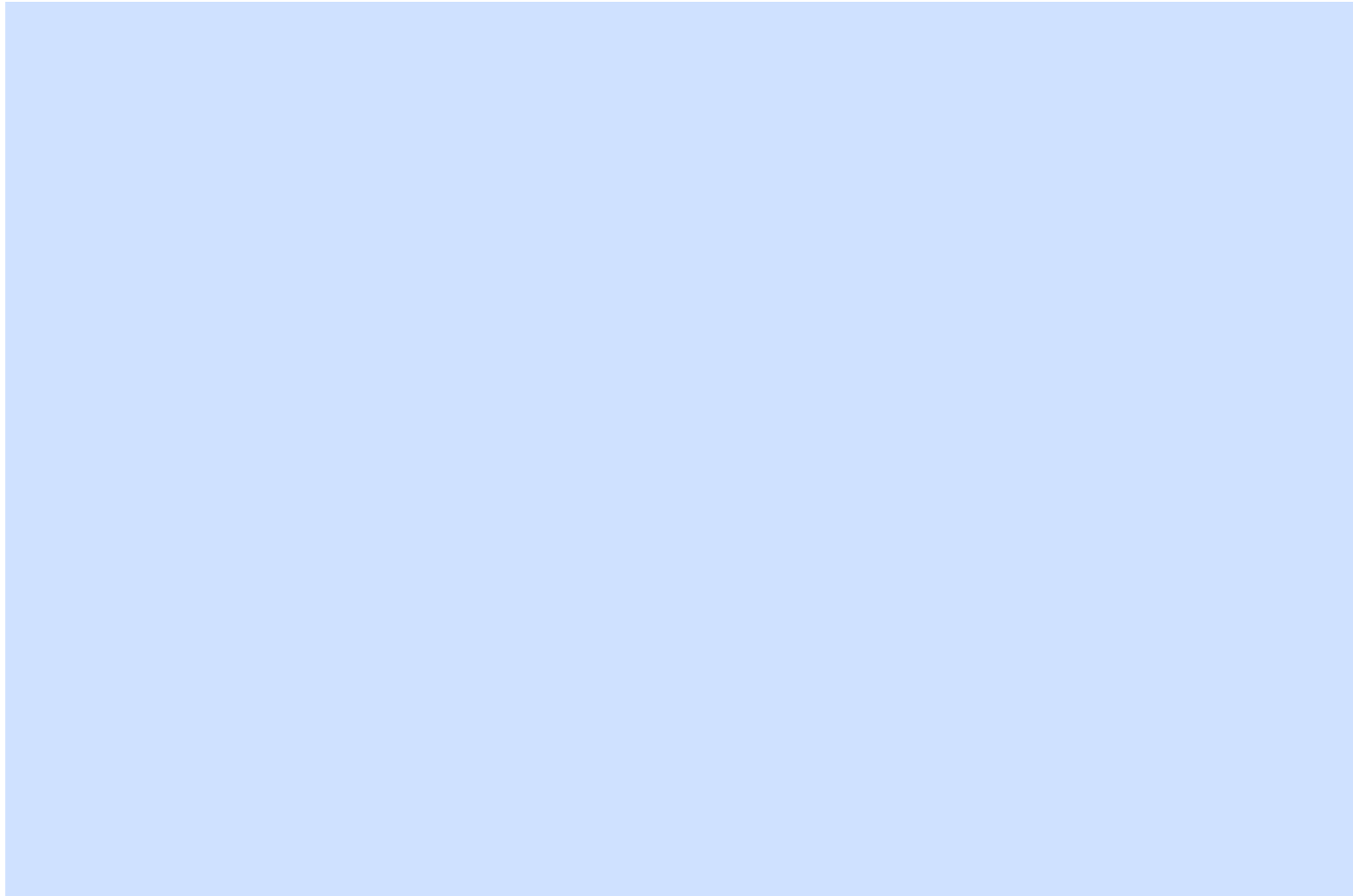


תכנות אקראי

- ▶ 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?



תכנות אקראי RANDOM

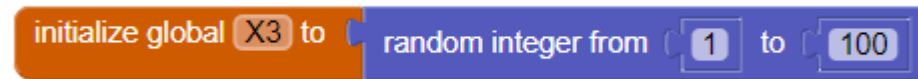
ספריית MATH

The image shows a Scratch interface. On the left is a library of blocks categorized into Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. The 'Math' category is highlighted. Below the library is a stage area with a 'Screen1' container holding 'Label1', 'Image1', and 'Label2'. A 'HorizontalArrangement' container holds 'MLKButton', 'MalcolmButton', and 'MLKPlayer'. The main script area contains several blue blocks: a multiplication block, a division block, an exponentiation block, a 'bitwise and' block, a 'random integer from 1 to 100' block, a 'random fraction' block, a 'random set seed to' block, and a 'min' block. A yellow 'when ML button clicked' block is partially visible behind the script.

אפשרות להגדלת מספר
מ 1 ל 100

תכנות אקראי RANDOM

למשל - מגרילים מספר רנדומלי בין 1 ל100 ומעדכנים את X3



initialize global X3 to random integer from 1 to 100

תרגיל מספר ■ - 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 מסתכרנות 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

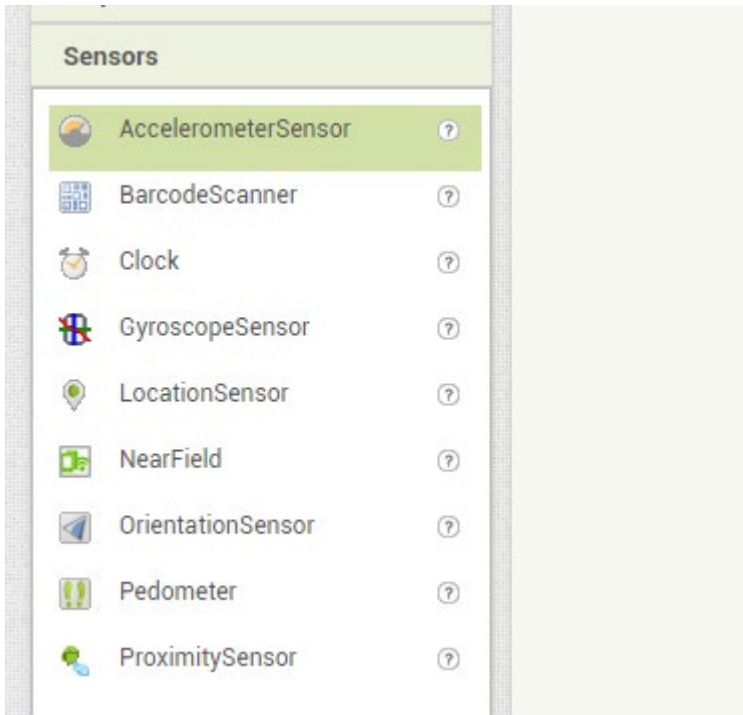
▶ סנסור (חיישן) תזוזה - Accelerometer

▶ סנסור סריקת ברקוד BARCODE SCANNER

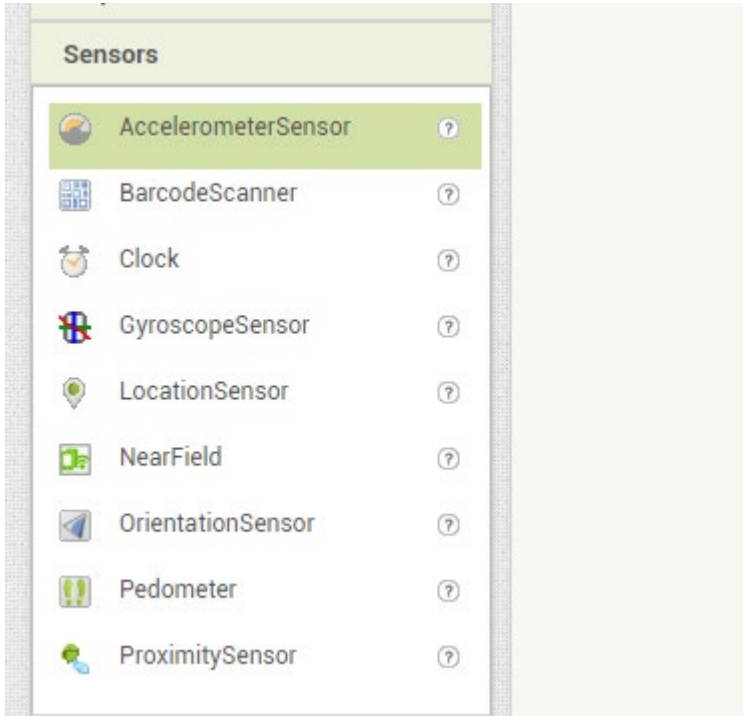
▶ סנסור GPS והמיקום LOCATION

▶ שעון - CLOCK

(מכשיר שאחראי על מדידת זמן בתוך המכשיר)



סנסורים



▶ חיישן GYROSCOPE -

האחראי על מהירות הסיבוב של המכשיר

▶ חיישן הכוון ORIENTATION -

חיישן המודד כוון של המכשיר מבחינת צפון ודרום

▶ חיישן הדיוק Proximity

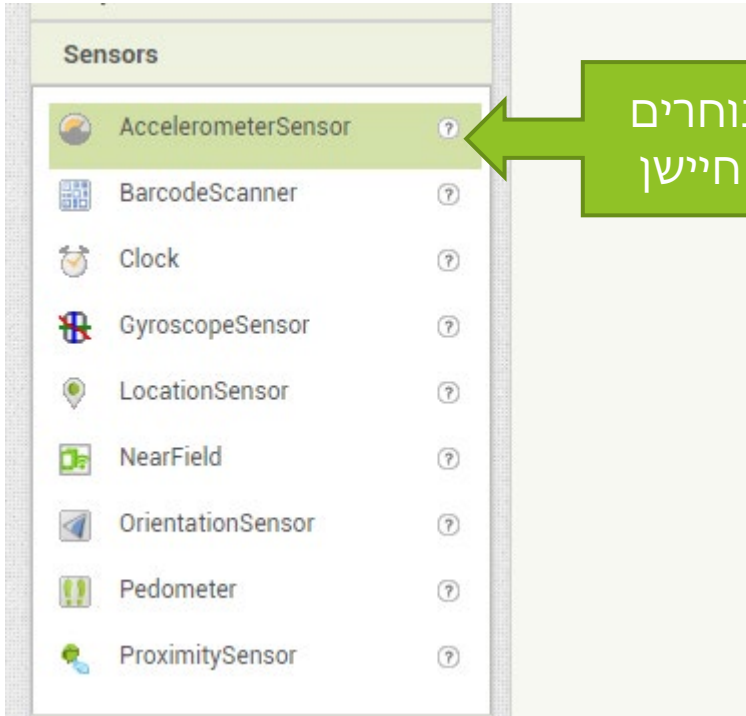
סוגי סנסורים

החיישנים ב-APP INVERTOR הם הרכיבים בלי נראים NON VISIBLE

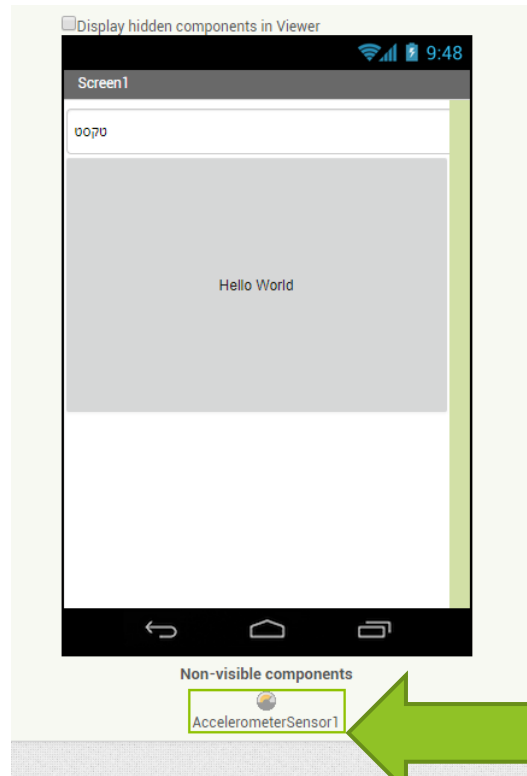
כאשר נגרור את החיישן למסך הוא יתוסף למטה מתחת למסך

משתמש לא יראה שאנחנו משתמשים בחיישן

אנחנו נקבל את אירועים של החיישן

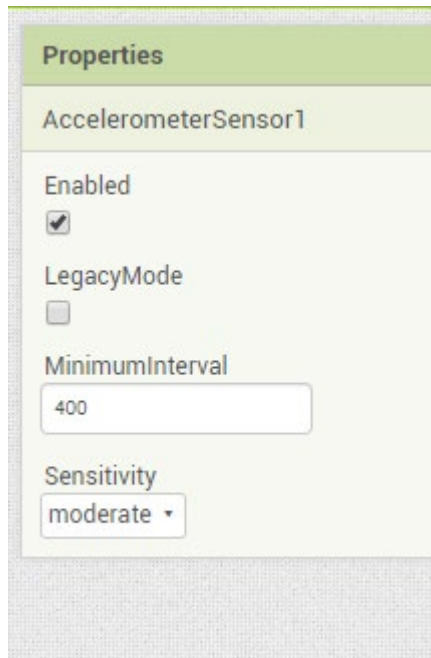


בוחרים
חיישן



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

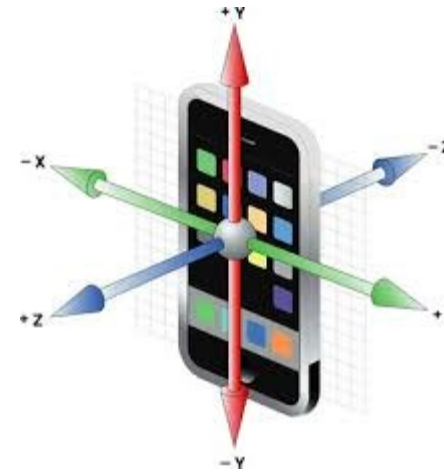
אירועים של ACCELEROMETER

The screenshot shows a visual programming interface with a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel lists various categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (TextBox1, Button1, AccelerometerSensor1), and Any component. The 'Viewer' panel displays a sequence of blocks for configuring 'AccelerometerSensor1':

- when AccelerometerSensor1 . AccelerationChanged do (containing 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

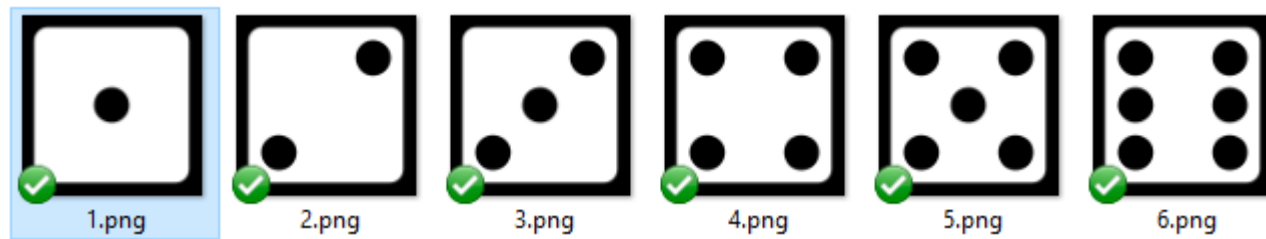


תרגיל מספר 1 - זריקת קובייה

- ▶ יש לפתח אפליקציה לזריקת הקובייה DICE
- ▶ הקובייה מכילה כיודע 6 צדדים.
- ▶ מעל כל צד מופיע מספרים מ 1 עד 6
- ▶ משתמש יכול להפעיל את האפליקציה באחד משני דרכים - לחיצה על כפתור למטה או נדנוד SHAKING של המכשיר

תרגיל מספר 1 - זריקת קובייה

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



תרגיל מספר 1 - זריקת קובייה

```
when RollButton.Click
do
  initialize local DiceRoll to random integer from 1 to 6
  in
    if get DiceRoll = 1
    then set Image1.Picture to "1.PNG"
    else if get DiceRoll = 2
    then set Image1.Picture to "2.PNG"
    else if get DiceRoll = 3
    then set Image1.Picture to "3.PNG"
    else if get DiceRoll = 4
    then set Image1.Picture to "4.PNG"
    else if get DiceRoll = 5
    then set Image1.Picture to "5.PNG"
    else if get DiceRoll = 6
    then set Image1.Picture to "6.PNG"
```

רשימה של רשימות LIST OF LISTS

- ▶ ב-APP INVERTOR ניתן לא רק לבנות LIST פשוט, אלה 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 ל-LIST 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



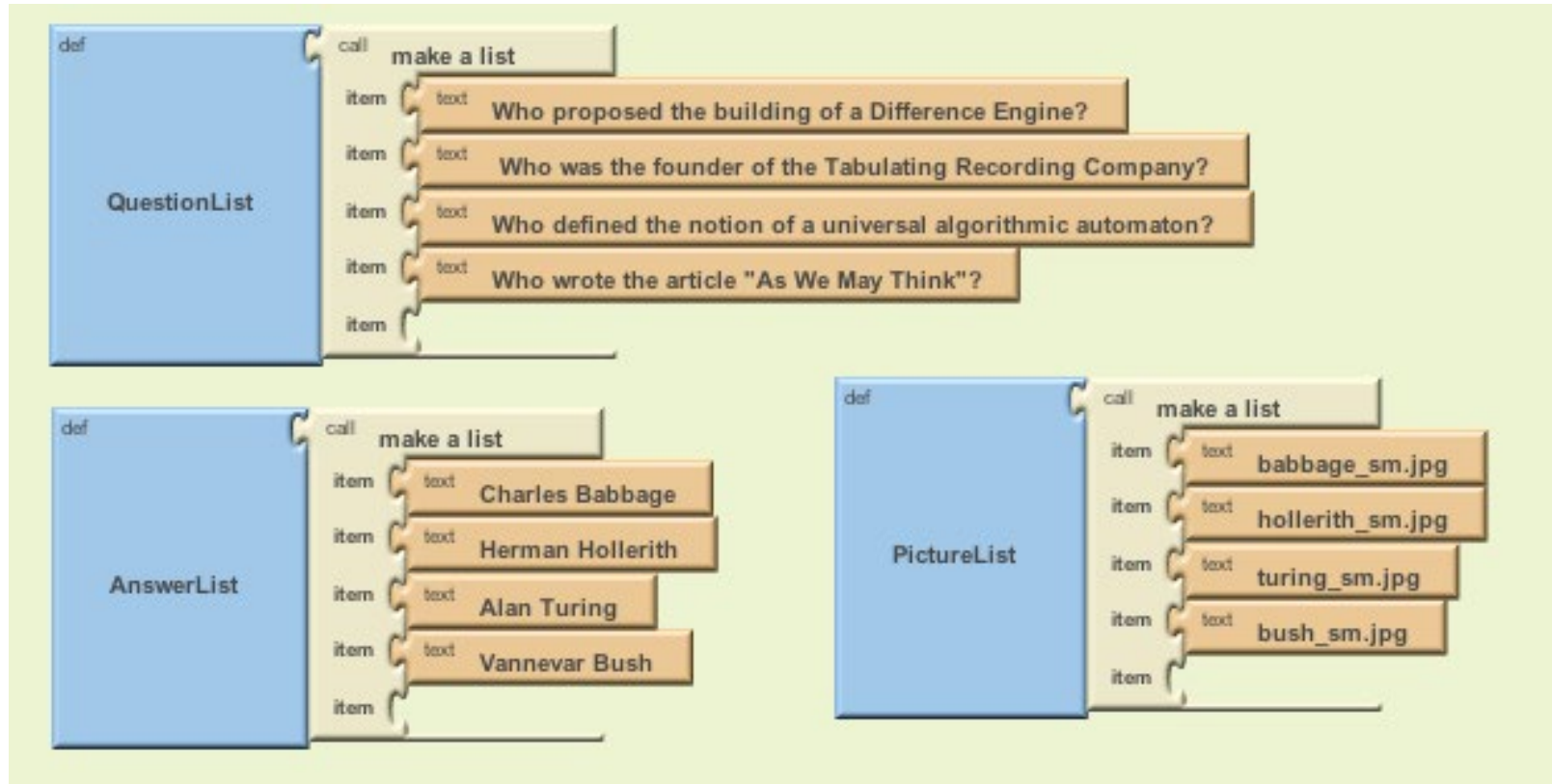
תרגיל מספר 4 - פרויקט משחק הטריז

- יש לפתח אפליקציה שמנהלת משחק טריז.
- 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 הטריז



תרגיל מספר 4 - ניתן למצוא דוגמא כאן -

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

פיתוח אפליקציות

שיעור מספר 8

שיעור ■ - גרפיקה ו-CANVAS

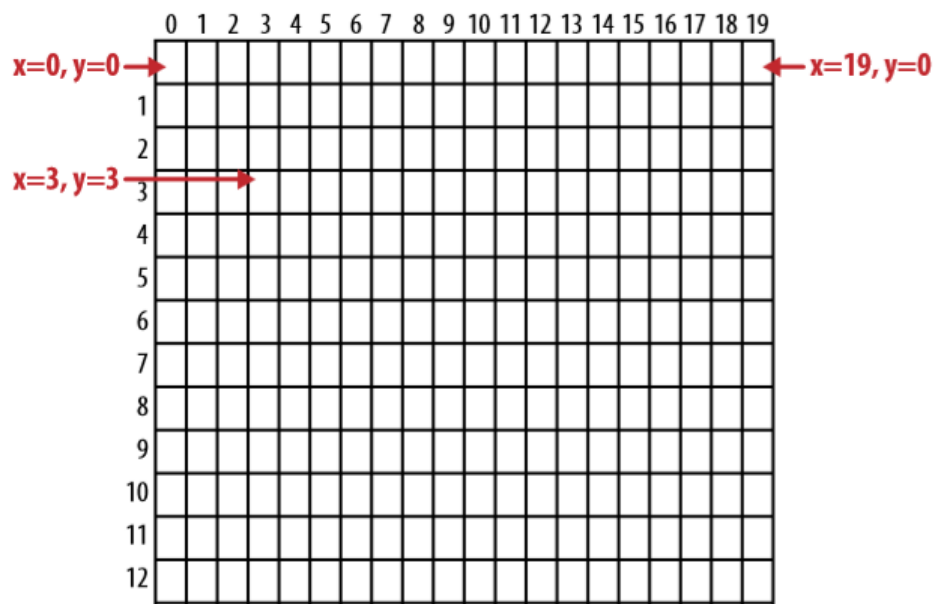
- .1 חזרה
- .2 CANVAS
- .3 בשיעור הזה נתרגל המון!!

תרגיל חזרה קטן

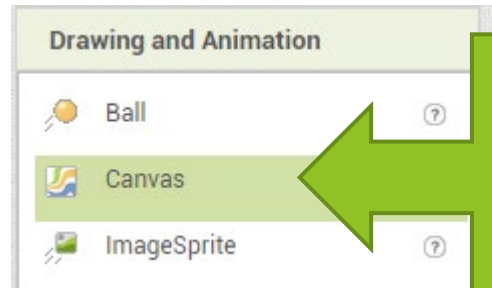
- ▶ יש לסיים את משחק הטרוויה שפיתחתם.
- ▶ במידה והמשחק הסתיים יש להוסיף תמיכה בהוספת שאלות נוספות
- ▶ תנסו לעצב יפה את צבע הרקע של השאלות והתשובות

CANVAS

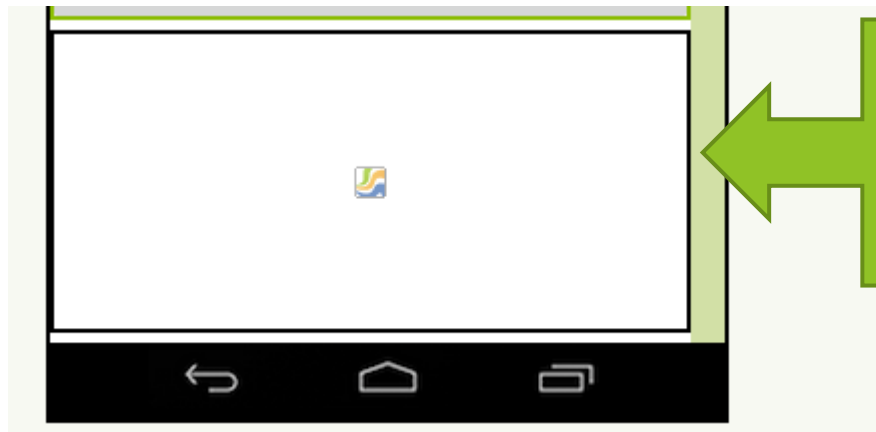
- ▶ על מנת לצייר, ולהציג ציורים וסרטים מצוירים באפליקציה, נדרשות יכולות גרפיות
- ▶ באנדרואיד, יכולות גרפיות ניתן להשיג באמצעות רכיב ה-CANVAS
- ▶ CANVAS הוא רכיב וניתן למקם אותו בכל מקום במסך.
- ▶ הרעיון הוא לחלק את מיקום הציור למערכת קואורדינטות בעלות כתובות X ו Y
- ▶ המיקום בפינה שמאלית העליונה מוגדר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



CANVAS



בחירת
CANVAS
תחת
DRAWING



מיקום
CANVAS
על גבי מסך

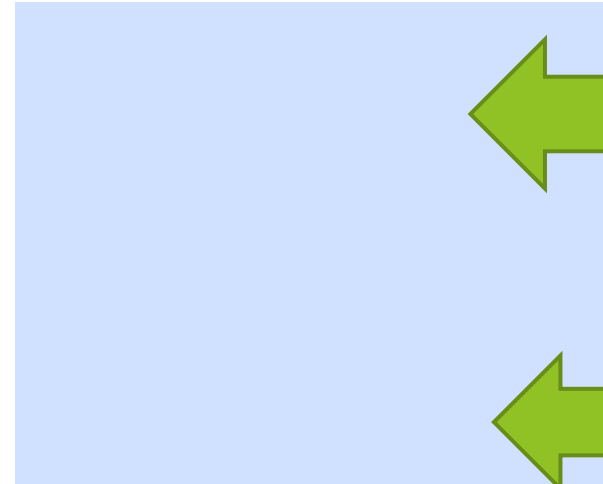


מומלץ להרחיב
את CANVAS
כך שיתפרס על
כל העמוד או
לפחות שיתפרס
לרוחב

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



ציור קו -
מעבירים X ו Y
עבור 2 נקודות
הסוף וההתחלה

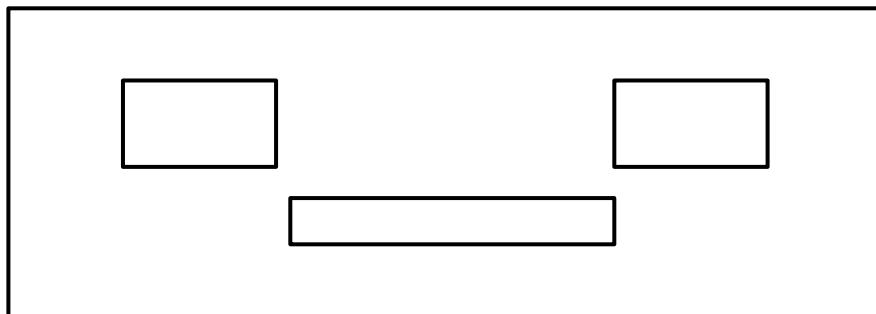
ציור של נקודה
מעבירים נ"צ
של הנקודה

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

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

תרגיל מספר

- 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

יש לצייר מספר קווים חשובים

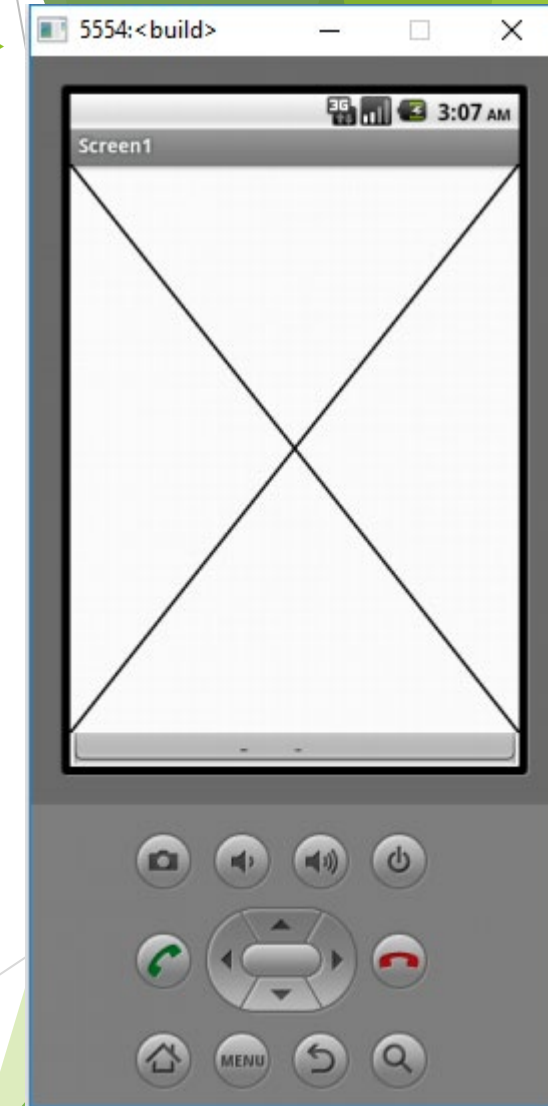
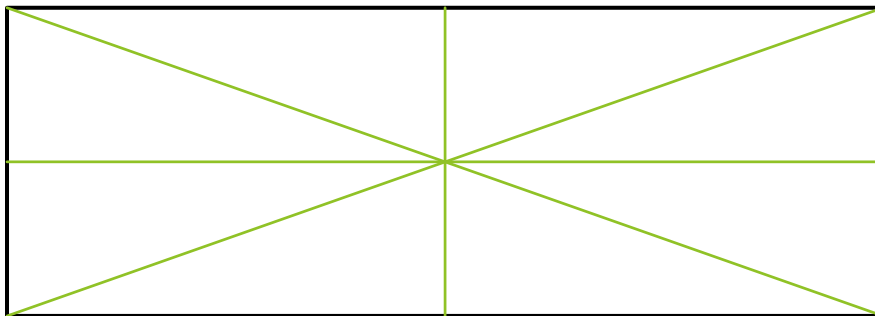
אלכסון ראשי

אלכסון משני

קו אמצע אנוכי

קו אמצע אופקי

שימו לב - הקווים צריכים להיפגש במרכז המסך



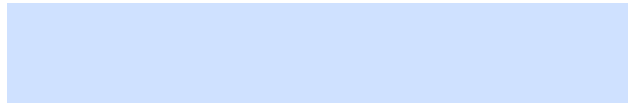
CANVAS

- ▶ 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

```
call Canvas1 .DrawShape  
pointList  
fill true
```

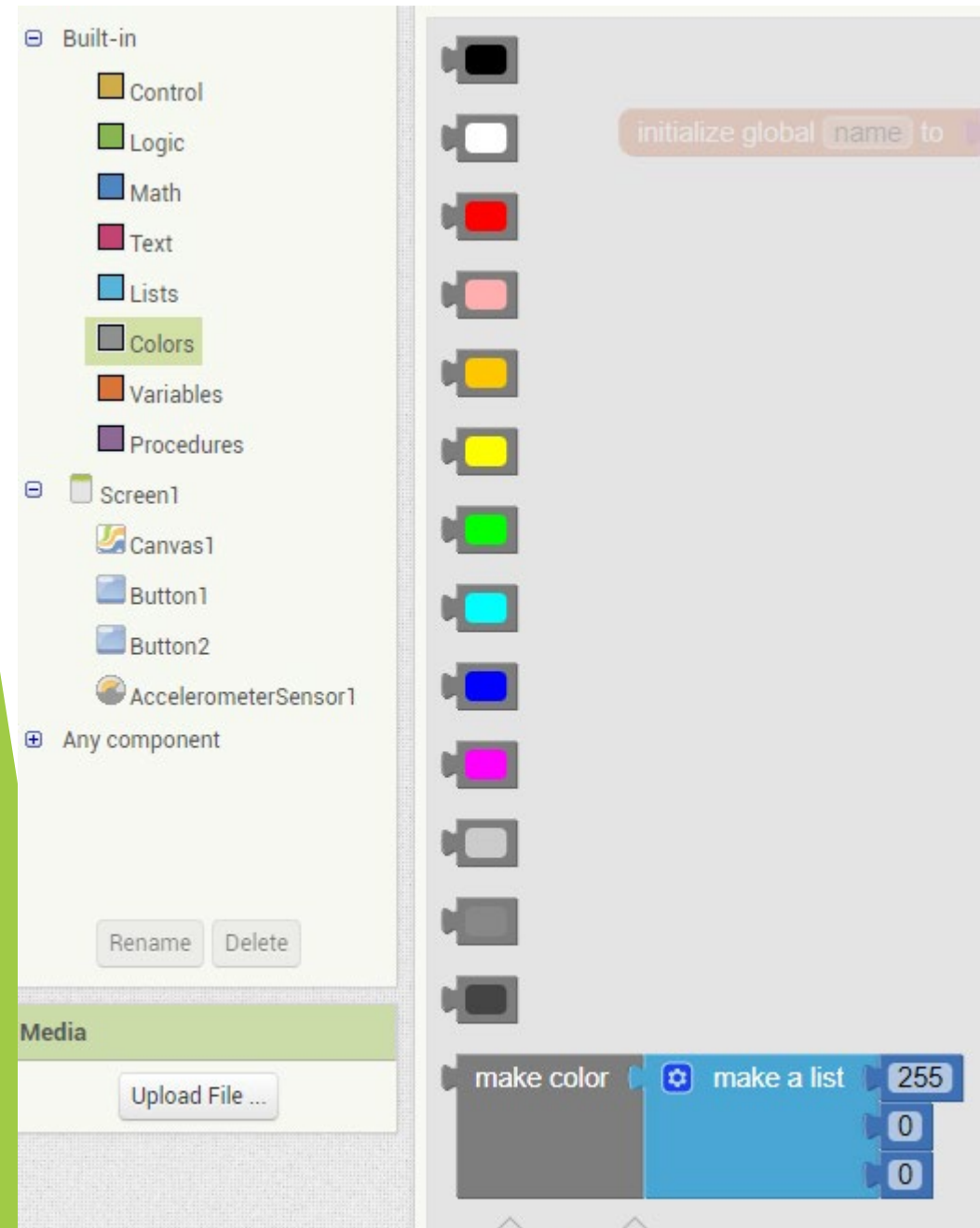
CANVAS

- ▶ 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

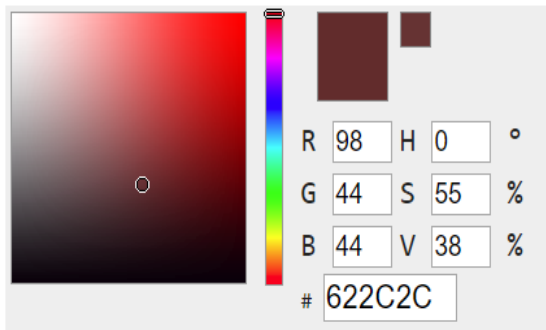
- ▶ ב APP INVERTOR ניתן לבחור צבעים קבועים או ניתן ליצור צבע חדש
- ▶ על מנת ליצור צבע חדש יש להקיש את מספרו תוך שימוש ב MAKE COLOR
- ▶ את בניית צבע ניתן לעשות דרך האינטרנט



RGB Color Codes Chart

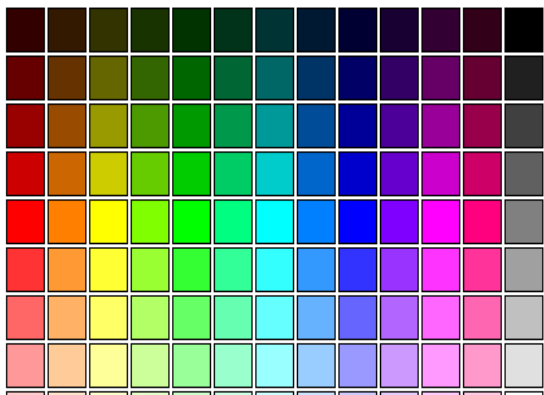
[RGB color picker](#) | [RGB color codes chart](#) | [RGB color space](#) | [RGB color format and calculation](#) | [RGB color table](#)

RGB color picker



RGB color codes chart

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



RGB עובד עם הצבעים בפורמט

RGB מסמל חיבור של 3 צבעים

אדום - RED

ירוק - GREEN

כחול -BLUE

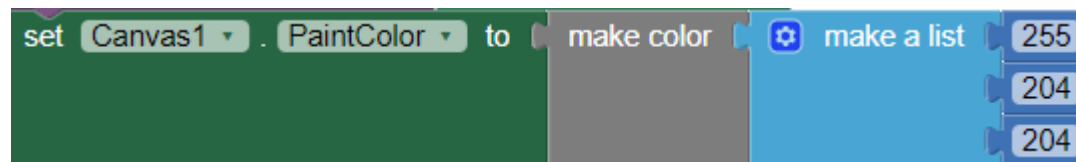
למשל באתר הזה ניתן לבנות עם הצבעים הללו כל צבע שתמצאו

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!!!!



תרגיל מספר 4

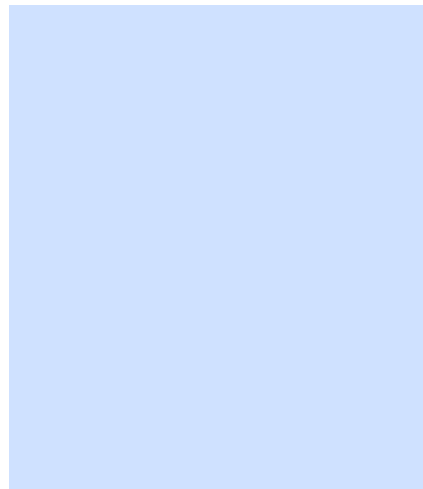
יש לצייר מגן דוד ▶

מגן דוד מורכב מ 2 משולשים ▶

יש לצייר אותו בצבע כחול ▶

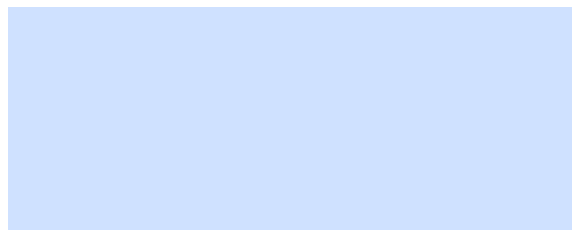
יש לקבוע את גודל הקו בהתאם ▶

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



```
when Button2 .Click  
do set global name to call Canvas1 .SaveAs  
fileName "aaa.jpeg"
```

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

```
when DrawingCanvas .Touched
  x y touchedAnySprite
do
  call DrawingCanvas .DrawCircle
    centerX get x
    centerY get y
    radius 5
    fill true
```

תרגיל מספר - ■ - 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

```
initialize global currentColor to make color make a list 255 204 204
```

- ▶ מעכשיו לפני שמציירים צורה מעדכנים את הצבע שלה

```
set Canvas1 PaintColor to get global currentColor
```



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

תרגיל מספר - ■ - ציור עיגולים בצבעים שונים

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

תרגיל מספר - ■ - 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

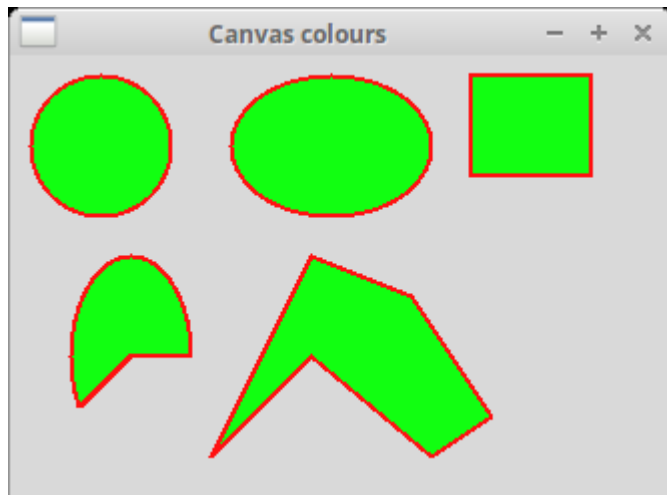
שיעור מספר 9

שיעור ■ - 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



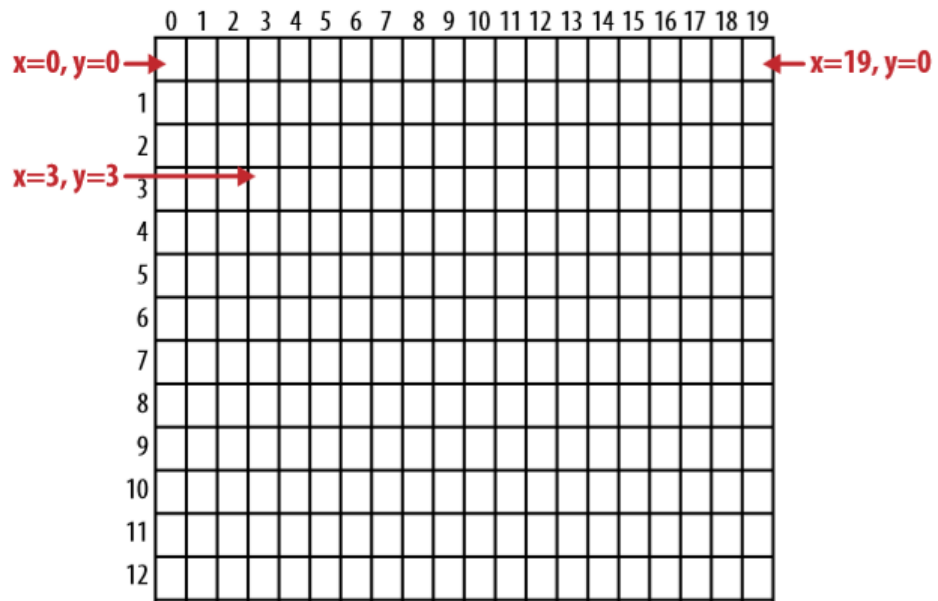
תרגיל מספר 1 - פרויקט ציור חופשי - PAINTPOT - שלב ב

- ▶ 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

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

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
```

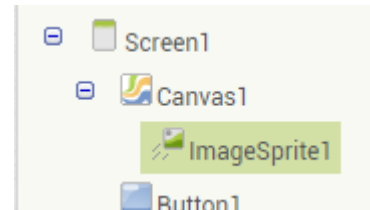
```
when DrawingCanvas .Touched
x y touchedAnySprite
do call DrawingCanvas .DrawCircle
  centerX get x
  centerY get y
  radius get global dotsize
  fill true

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 ה-CANVAS. SPRITE יכול לשתף פעולה עם SPRITE ים קיימים אחרים על גבי אותו CANVAS



SPRITE - CANVAS

Canvas1

- ImageSprite1
- Button1
- Button2
- AccelerometerSensor1

Enabled

Heading

Height

Width

Interval

Picture

Rotates

Speed

Visible

X

Y

Z

Rename Delete

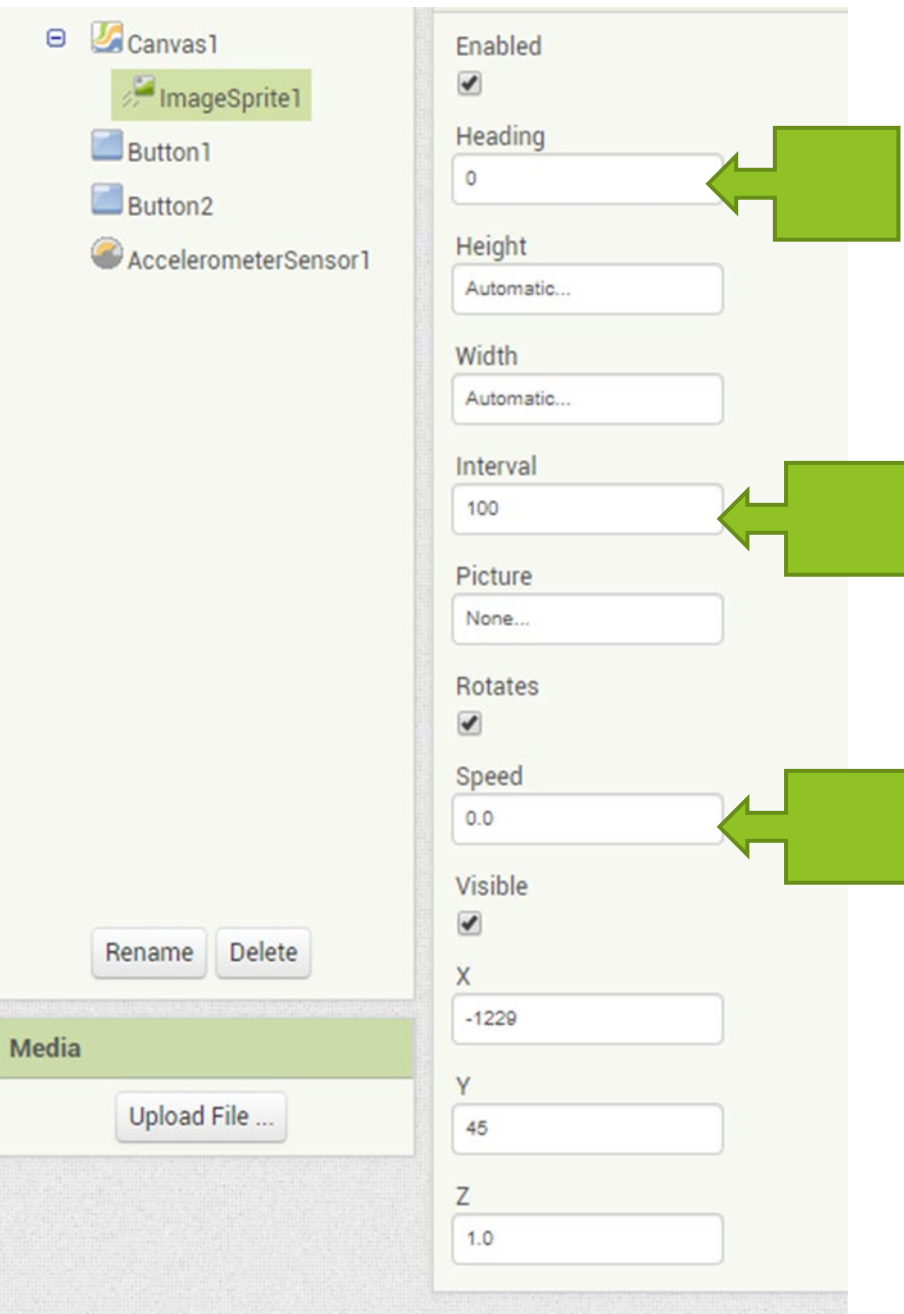
Media

Upload File ...

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

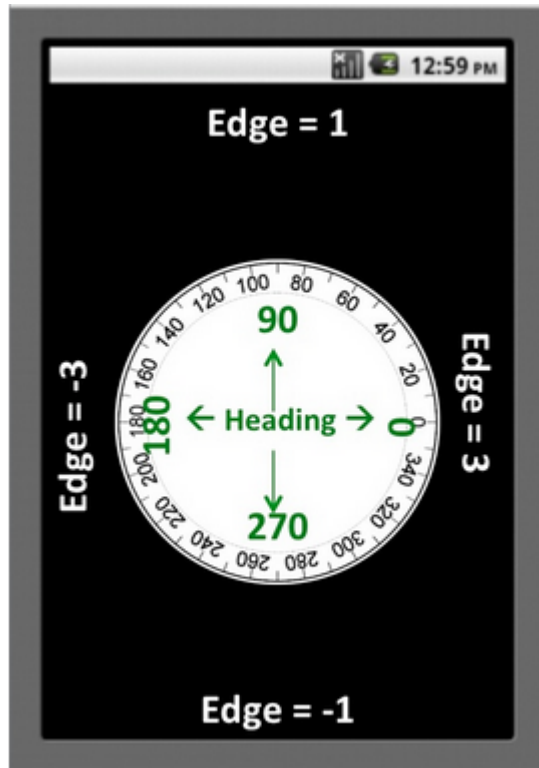
מרווח - INTERVAL

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

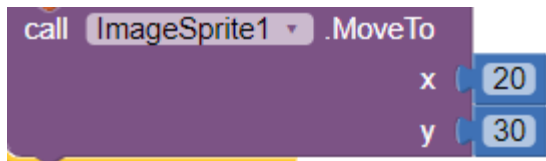
ROTATES קובע האם הצורה זזה לפי הכוון שהוגדר לה בHEADINGS

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



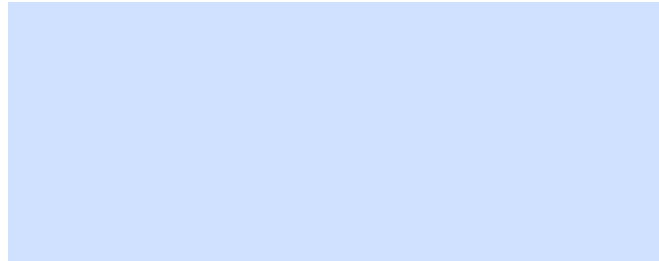
■ תרגיל מספר

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



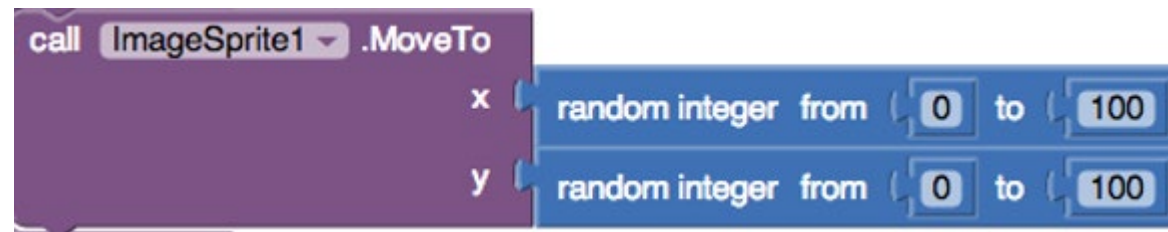
SPRITE - אינטרקציה עם דמויות אחרות

- ▶ לכל SPRITE ניתן לבדוק האם הוא מתנגש או חוצה את הדרך של SPRITE אחר כלשהו
- ▶ לצורך כך קיימים לSPRITE אירועים נוספים
- ▶ אירוע COLLEIDEDWITH - APP INVERTOR 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



```
call ImageSprite1 .MoveTo
  x random integer from 0 to 100
  y random integer from 0 to 100
```

The image shows a Scratch code block for moving a sprite. The block is purple and has a dropdown menu set to 'ImageSprite1'. The method name is '.MoveTo'. There are two input fields: 'x' and 'y'. Each field is connected to a blue 'random integer from 0 to 100' block.

תרגיל מספר 4 - התנגשות בין ה-MOLEים

- 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

```
to moveMole
do call ImageSprite1 .MoveTo
  x random integer from 0 to 100
  y random integer from 0 to 100
end
```


■ תרגיל מספר

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

```
when Mole .Touched
do
  set global score to [get global score] + 1
  call Noise .Vibrate
  milliseconds 100
  call UpdateScore
  call MoveMole
```

פרויקט - תרגיל מספר

- 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/molemash.html>

```
to MoveMole
do
  set Mole . X to random fraction × (MyCanvas . Width - Mole . Width)
  set Mole . Y to random fraction × (MyCanvas . Height - Mole . Height)
```

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

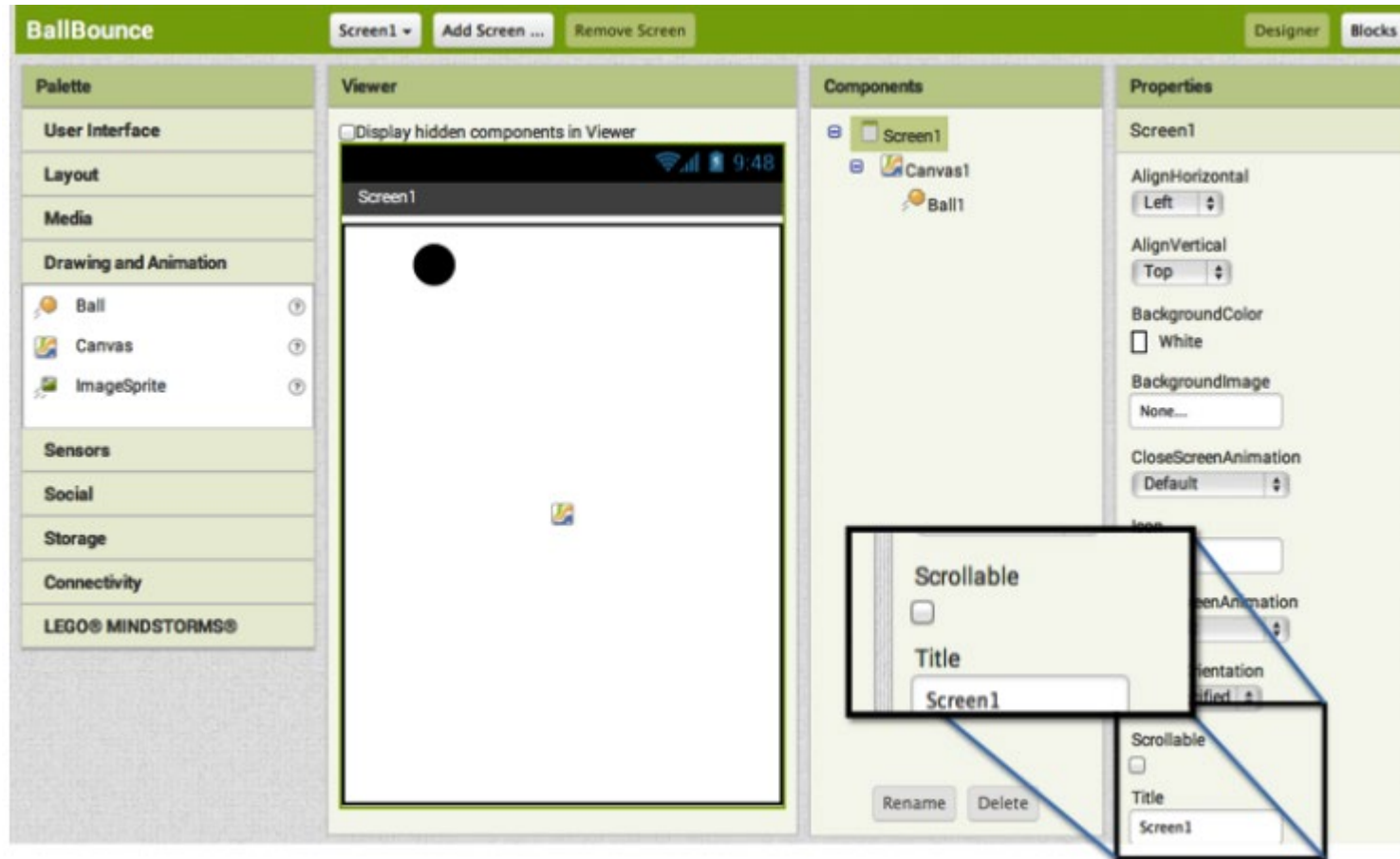
- יש לעבור על התרגילים שלמדנו בשיעור הקודם
- יש לבדוק שפרויקט MOVEMOL עובד

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

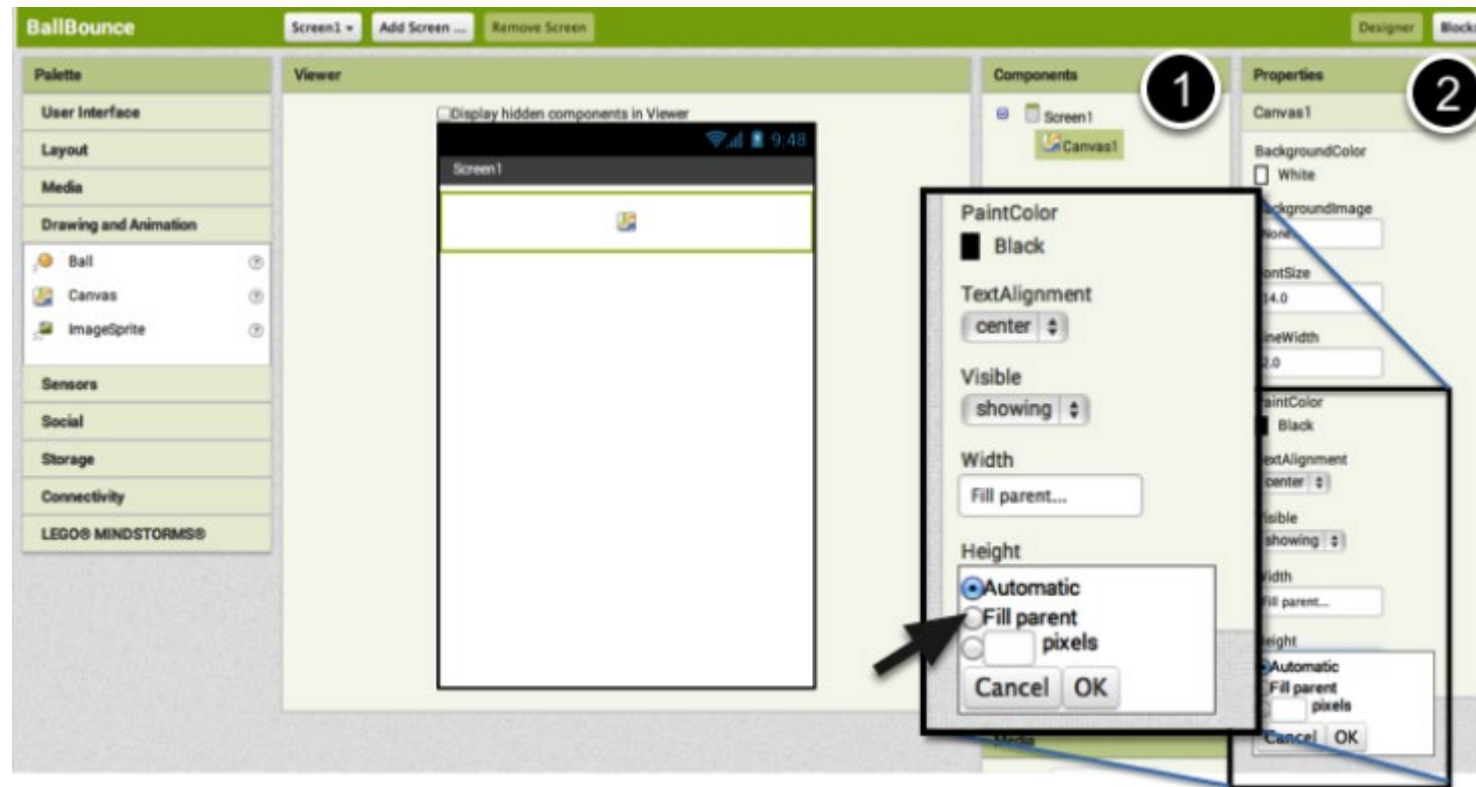
פיתוח תנועת כדור

- ▶ 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

The screenshot displays the LEGO Mindstorms software interface with four main panels:

- Palette:** The 'Media' category is selected, and the 'Ball' component is highlighted with a red circle and a '1' in a black circle. An orange arrow points from the 'Ball' icon to the viewer.
- Viewer:** A black circle representing the ball is placed on the canvas. A red circle and an orange arrow highlight the ball.
- Components:** The 'Ball1' component is listed under 'Canvas1'.
- Properties:** The 'Ball1' properties are shown. The 'Radius' property is set to 15, highlighted with a red circle and a '2' in a black circle. Other properties include 'PaintColor' (Black), 'Interval' (100), 'X' (48), and 'Y' (11).

פיתוח תנועת כדור

לBALL יש מספר פעולות

The image shows a Scratch code editor with a 'Blocks' panel on the left and a 'Viewer' panel on the right. The 'Blocks' panel has a 'Ball1' block circled in red. The 'Viewer' panel shows several code blocks for the Ball1 object, each with a green callout box pointing to it:

- when Ball1 .CollidedWith other**: אובייקט אחר בדיקת התנגשות עם
- when Ball1 .Dragged** (with parameters: startX, startY, prevX, prevY, currentX, currentY): גרירת כדור
- when Ball1 .EdgeReached edge**: בדיקת התנגשות עם קצה CANVAS
- when Ball1 .Flung** (with parameters: x, y, speed, heading, xvel, yvel):
- when Ball1 .No longerCollidingWith**: (partially visible)

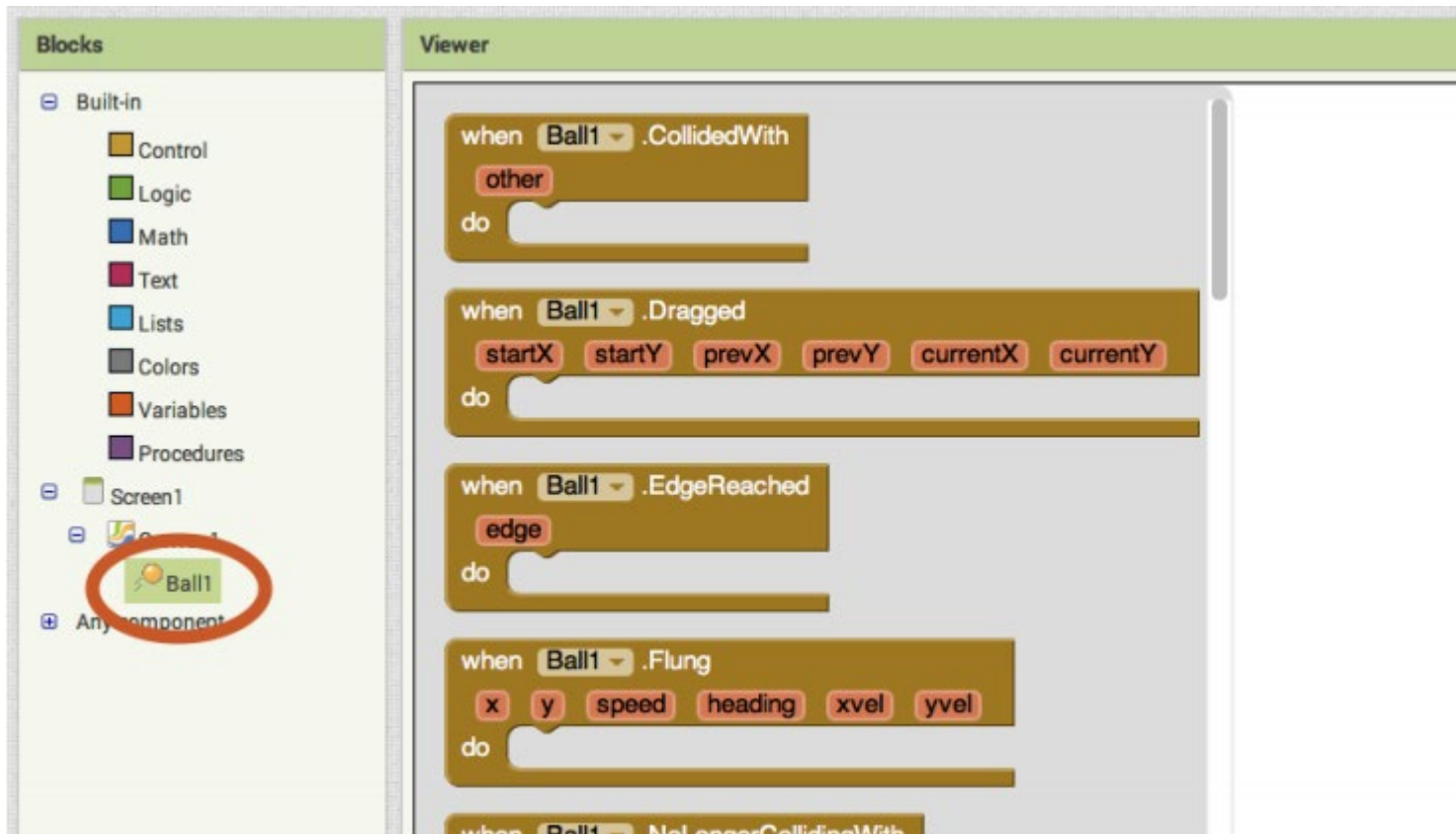
פיתוח תנועת כדור FLUNG

- ▶ 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

```
when Ball1 .Flung
  x y speed heading xvel yvel
do
```

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

The image shows a programming environment interface. On the left is the 'Blocks' palette, and on the right is the 'Viewer' window. The 'Blocks' palette is organized into categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1, Canvas1, and Any Component. The 'Ball1' component is selected and circled in red. The 'Viewer' window displays a sequence of blocks for 'Ball1'. The blocks are: 'set Ball1 . Enabled to', 'Ball1 . Heading', 'set Ball1 . Heading to', 'Ball1 . Interval', 'set Ball1 . Interval to', 'Ball1 . PaintColor', 'set Ball1 . PaintColor to', 'Ball1 . Radius', 'set Ball1 . Radius to', 'Ball1 . Speed', 'set Ball1 . Speed to', and 'Ball1 . Visible'. The 'set Ball1 . Heading to' and 'set Ball1 . Speed to' blocks are circled in red.

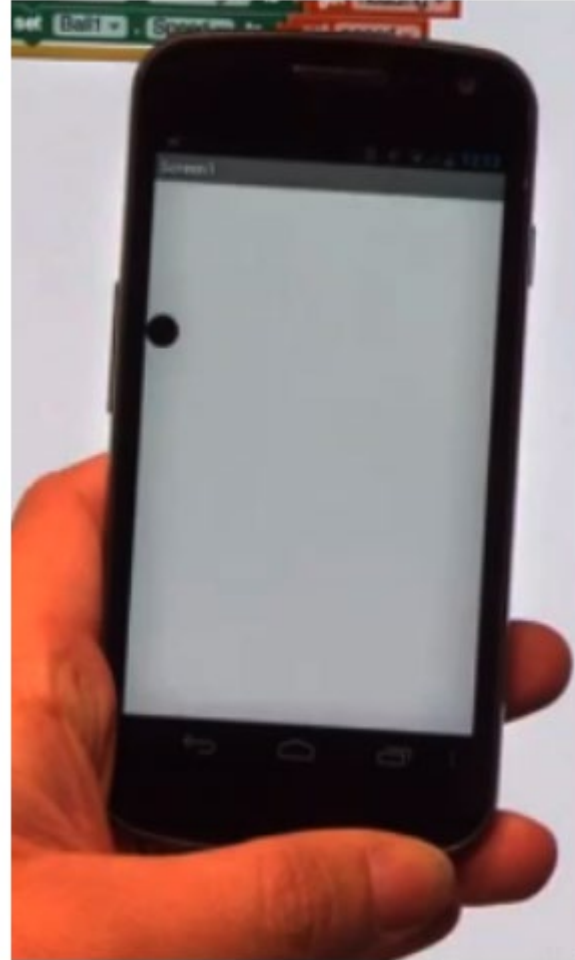
פיתוח תנועת כדור

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

```
when Ball1 . Flung
  x y speed heading xvel yvel
do
  set Ball1 . Speed to get speed
  set Ball1 . Heading to get heading
```

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

The screenshot displays a programming environment with two main panels: 'Blocks' on the left and 'Viewer' on the right. In the 'Blocks' panel, the 'Ball1' component is selected and circled in red. The 'Viewer' panel shows a script for 'Ball1' with several event-based blocks:

- when Ball1 .CollidedWith** (with an 'other' block in the 'do' field)
- when Ball1 .Dragged** (with variables: startX, startY, prevX, prevY, currentX, currentY in the 'do' field)
- when Ball1 .EdgeReached** (with an 'edge' block in the 'do' field, circled in red with an arrow pointing to the right)
- when Ball1 .Moving** (with variables: x, y, speed, heading, xvel, yvel in the 'do' field)
- when Ball1 .NoLongerCollidingWith** (with an 'other' block in the 'do' field)

At the bottom of the 'Blocks' panel, there are 'Rename' and 'Delete' buttons.

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

The image shows a programming environment with a 'Blocks' panel on the left and a 'Viewer' panel on the right. In the 'Blocks' panel, the 'Ball1' block is circled in red. In the 'Viewer' panel, several code blocks are visible, with the 'when Ball1 .EdgeReached' block also circled in red. An orange arrow points from this circled block to the right. The code blocks in the viewer include:

- when Ball1 .CollidedWith
other
do
- when Ball1 .Dragged
startX startY prevX prevY currentX currentY
do
- when Ball1 .EdgeReached
edge
do
- when Ball1 .Moving
x y speed heading xvel yvel
do
- when Ball1 .NoLongerCollidingWith
other
do

פיתוח תנועת כדור

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

The screenshot displays a visual programming interface with two main panels: 'Blocks' on the left and 'Viewer' on the right. The 'Blocks' panel lists various categories such as 'Built-in', 'Control', 'Logic', 'Math', 'Text', 'Lists', 'Colors', 'Variables', and 'Procedures'. Under 'Built-in', there are sub-categories like 'Screen1', 'Canvas1', and 'Ball1'. The 'Viewer' panel shows a sequence of code blocks for a ball object named 'Ball1'. The blocks include:

- when Ball1 .TouchDown (with x and y inputs) and a do block.
- when Ball1 .TouchUp (with x and y inputs) and a do block.
- when Ball1 .Touched (with x and y inputs) and a do block.
- when Ball1 .EdgeReached (with an edge input) and a do block.
- call Ball1 .Bounce (with an edge input) - this block is circled in red, and an orange arrow points from it to the 'EdgeReached' block.
- call Ball1 .CollidingWith (with an other input).

In the background, a zoomed-in view of the 'EdgeReached' block is visible, showing its 'do' block containing 'set Ball1 .Speed to get' and 'set Ball1 .Heading to get'.

פיתוח תנועת כדור

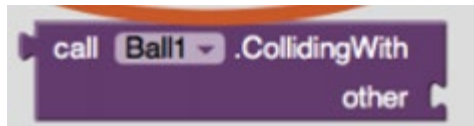
▶ BALL BOUNCE מקבלת כפרמטר EDGE הקצה ובעצם מבצע סיבוב של הכדור בהתאם למיקומו

▶ עכשיו יש להפעיל אפליקציה ולעקוב אחרי הכדור

```
when Ball1 .EdgeReached
  edge
do call Ball1 .Bounce
  edge get edge
```

תרגיל - פיתוח שלב 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

```
when PhoneCall1 .IncomingCallAnswered
  phoneNumber
do call PhoneCall1 .MakePhoneCall

when PhoneCall1 .PhoneCallEnded
  status phoneNumber
do

when PhoneCall1 .PhoneCallStarted
  status phoneNumber
do

call PhoneCall1 .MakePhoneCall
call PhoneCall1 .MakePhoneCallDirect

PhoneCall1 . PhoneNumber
set PhoneCall1 . PhoneNumber to
PhoneCall1
```

נקרא כאשר מגיע שיחה נכנסת

נקרא כאשר שיחה מסתיימת

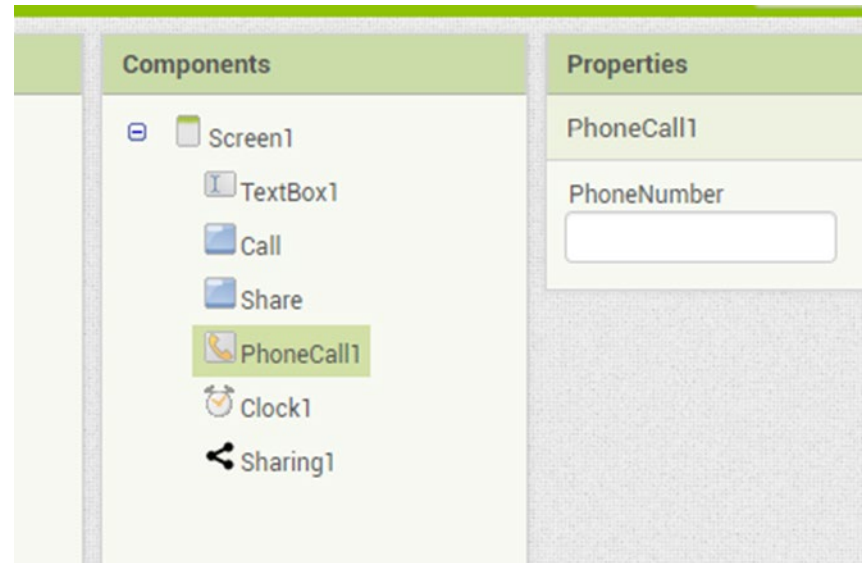
נקרא כאשר שיחה מתחילה

מחייג על פי מספר טלפון

משפיע - מחזיר או משתנה את מספר טלפון

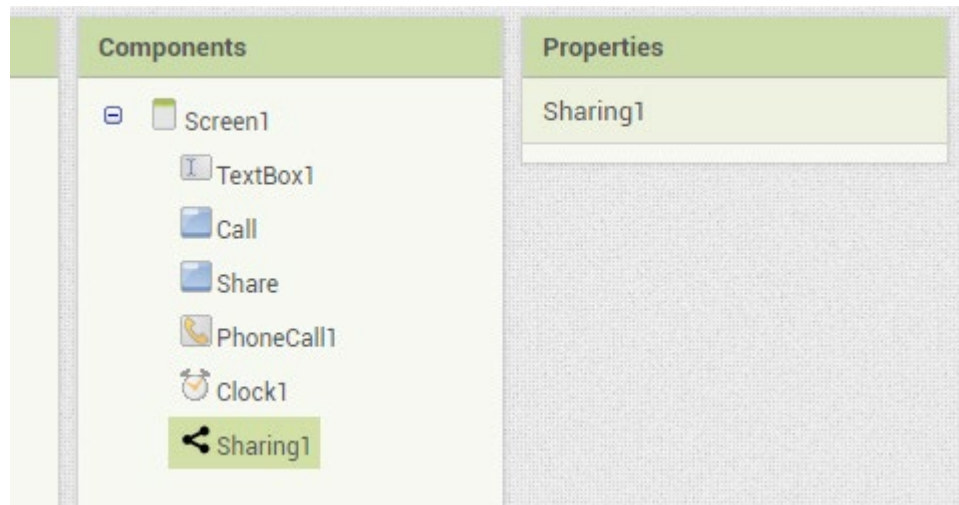
ביצוע שיחת טלפון וכפתור SHARING

בבחירת PHONECALL יש רק מאפיין אחד - מספר טלפון ►



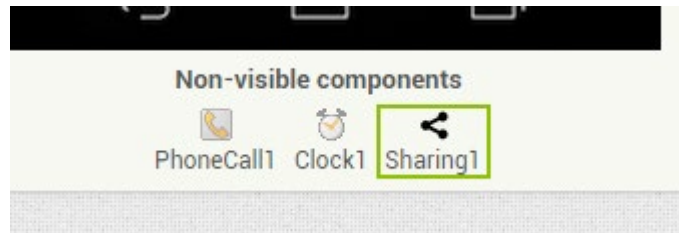
ביצוע שיחת טלפון וכפתור SHARING

- ▶ 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

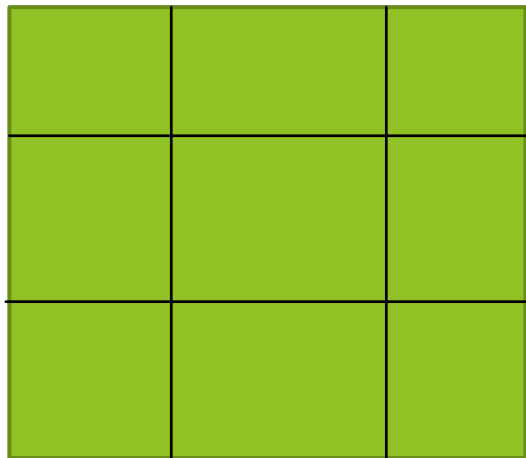
- ▶ כתוב אפליקציה, המבצעת שיחת טלפון
- ▶ משתמש יכול להזין מספר טלפון וללחוץ על הכפתור OK לביצוע השיחה
- ▶ בנוסף כאשר מקבל משתמש שיחה נכנסת, האפליקציה מדפיסה את מספר הטלפון הנכנס
- ▶ בנוסף אפליקציה תבצע SHARE לטלפון שהתקבל (למשל לתוך ה WHATSAPP)

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

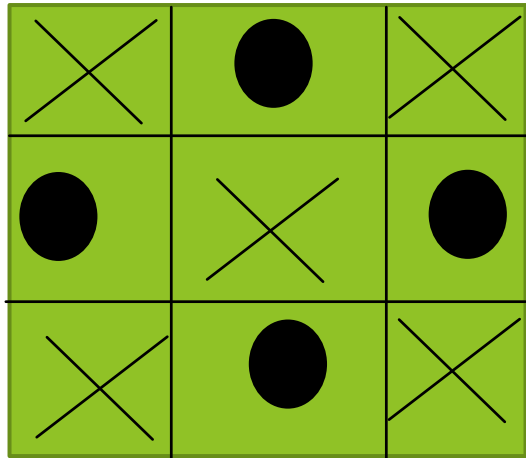
תרגיל מספר 1 יש לפתח אפליקציה למשחק איכס עיגול

יש קודם כל לצייר לוח עם 9 ריבועים



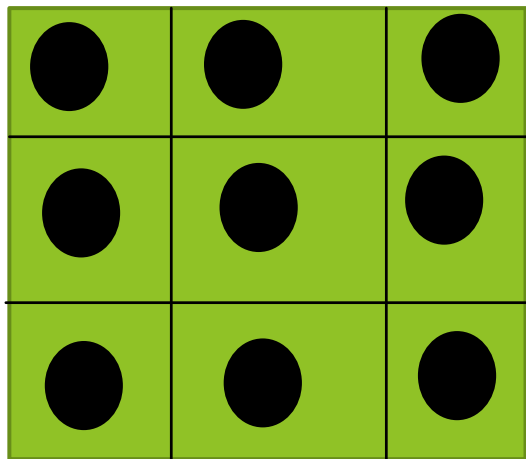
יש לפתח אפליקציה למשחק איכס עיגול

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



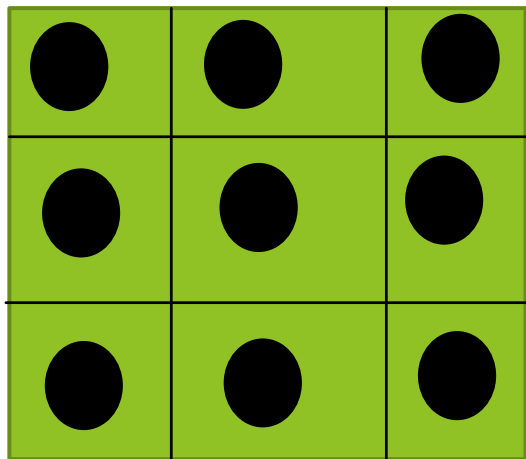
פרויקט אופציונלי - מורכב - טיק טק טו

- 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

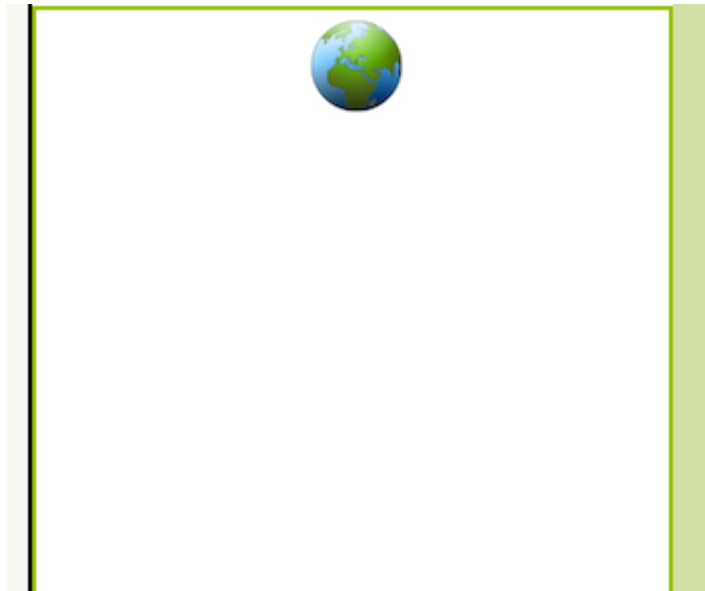
- ▶ יש להוסיף לתרגיל הזת כדור, שפיתחנו בשיעור הקודם, אפשרות להתנגש עם הצורות האחרות
- ▶ יש להוסיף עוד 5 צורות אחרות על הגבי המסך
- ▶ יש לאפשר התנגשות בין צורות - במידה והצורות מתנגשות יש לשנות את כוון התנועות

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

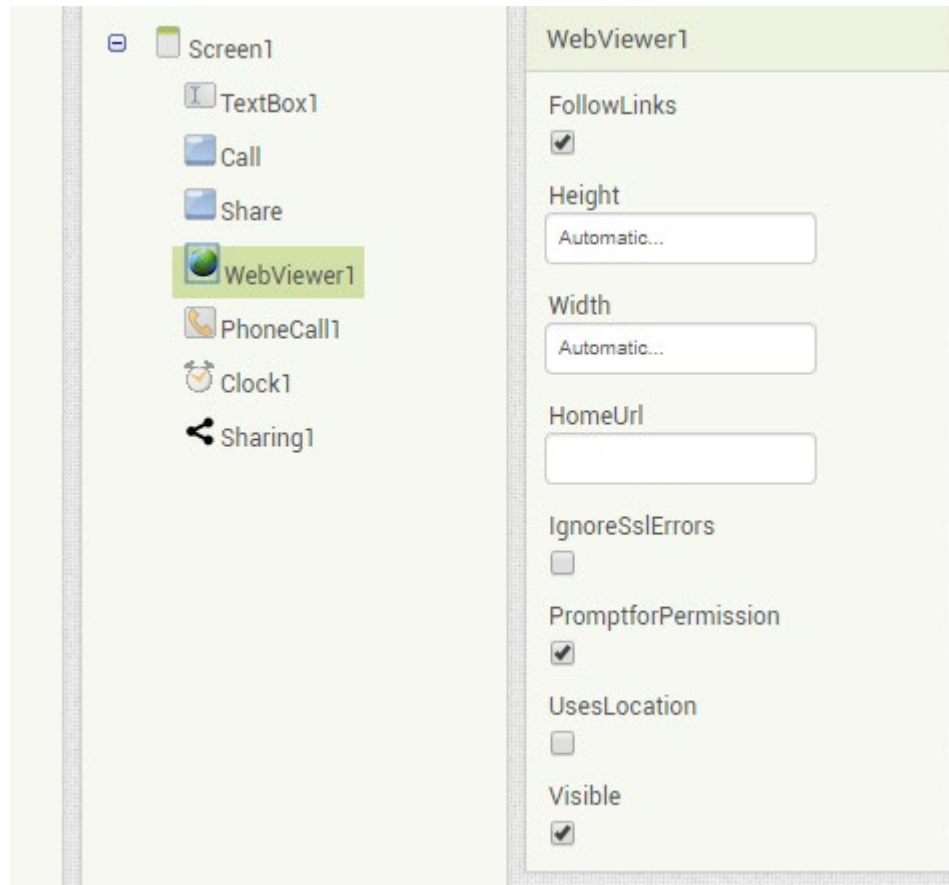
- ▶ האפליקציה שלנו יכולה להציג גם דפדפן או אתר כלשהו שפיתחנו
- ▶ המשתמש לא תמיד יכול ומסוגל להבחין בכך.
- ▶ למשל, האם הבחנתם שYNET או WALLA הם בעצם אתרים מוסתרים בדפדפנים.
- ▶ דרך פיתוח זאת מקלה מאוד על החברות - הם לא יצטרכו לפתח בו זמנים אפליקציה וWEB



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

```
when WebView1 .WebViewStringChange
do
call WebView1 .CanGoBack
call WebView1 .CanGoForward
call WebView1 .ClearCaches
call WebView1 .ClearLocations
call WebView1 .GoBack
call WebView1 .GoForward
call WebView1 .GoHome
call WebView1 .GoToUrl
url
WebView1 .CurrentPageTitle
WebView1 .CurrentUrl
WebView1 .FollowLinks
set WebView1 .FollowLinks to
WebView1 .Height
Show Warnings
set WebView1 .Height to
```

■ תרגיל מספר

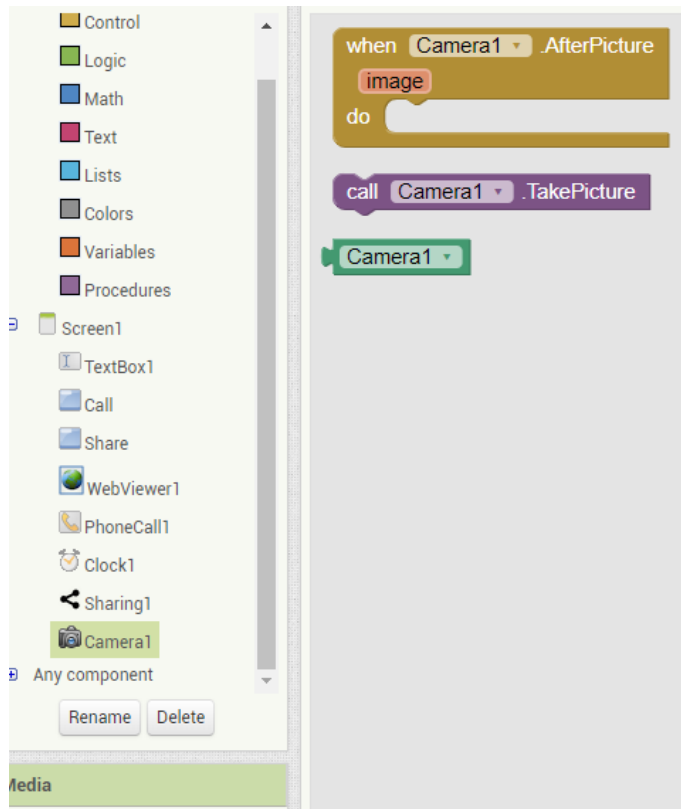
- ▶ פתח אפליקציה המציגה אתר YNET
- ▶ ובלחיצה על כפתור עוברת ומציגה אתר WALLA
- ▶ ובלחיצה על עוד כפתור מציגה אתר CNN
- ▶ שימו לב - על מנת להציג אתר יש לכתוב כתובת האתר בפורמט
- ▶ <http://www.ynet.co.il>

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

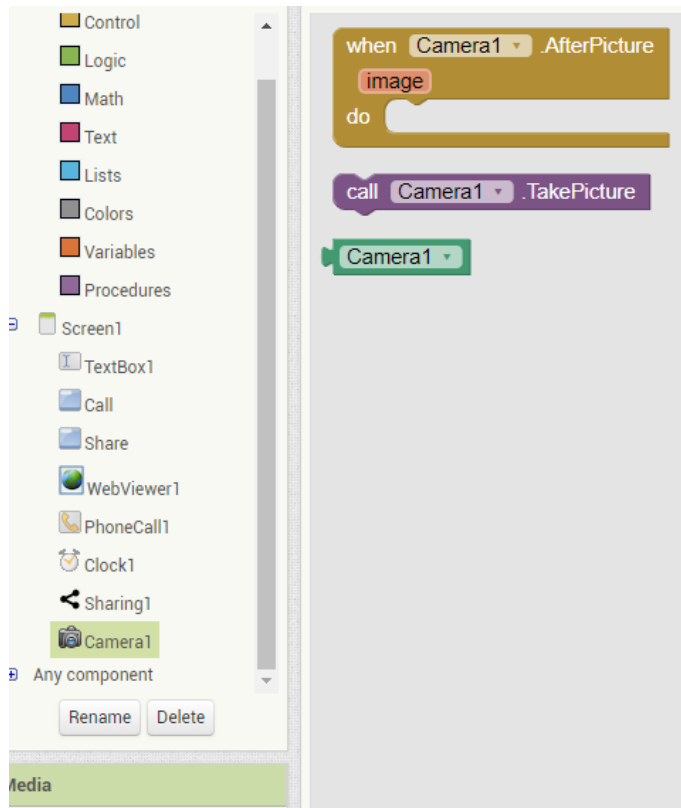
מצלמה ושימוש במצלמה

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



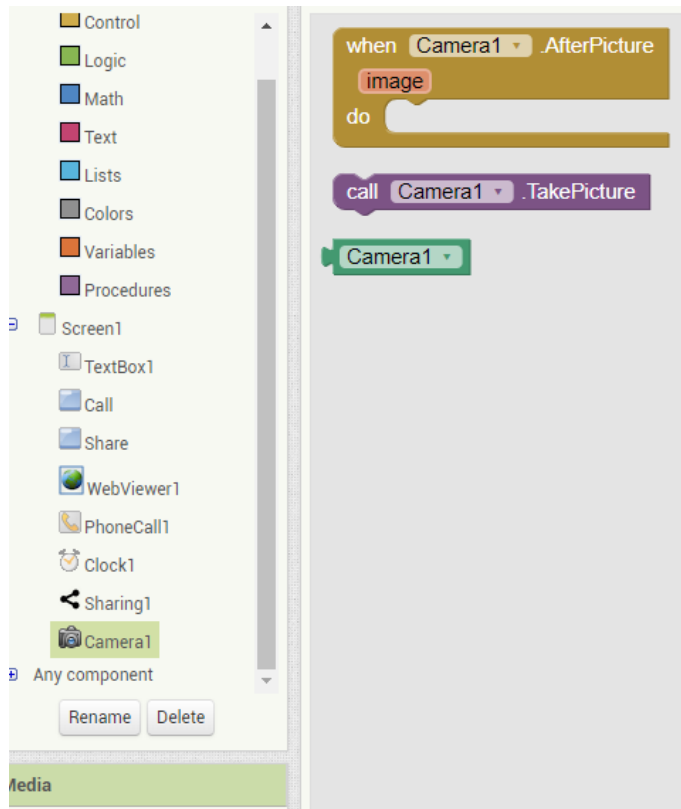
תרגיל מספר 1 - מלצמה ושימוש בה

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



תרגיל מספר 1 - 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

The image shows the MIT App Inventor web interface. At the top, the MIT App Inventor logo is on the left, and navigation menus for 'Projects', 'Connect', 'Build', and 'Help' are on the right. Below the header, the app name 'CallPhone' is displayed in a green bar, along with a dropdown menu for 'Screen1' and an 'Add Screen ...' button. A white tooltip box is positioned over the 'Add Screen ...' button, containing two options: 'App (provide QR code for .apk)' and 'App (save .apk to my computer)'. The main workspace is divided into two panels: 'Palette' on the left and 'Viewer' on the right. The 'Palette' panel is titled 'User Interface' and lists various UI components with their respective icons and a help icon (?): Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. The 'Viewer' panel shows a preview of the mobile app. At the top of the preview is a status bar with icons for Wi-Fi, signal strength, battery, and the time '9:48'. Below the status bar is a dark header labeled 'Screen1'. The main content area of the app contains a white text input field, a 'Call' button, and a 'Share...' button. At the bottom of the app preview is a large rectangular area containing a globe icon, which is likely a WebViewer component. The bottom of the preview shows a portion of the Android navigation bar.

יש לפתוח חוצץ BUILD ולבחור אופציה SAVE APK TO MY COMPUTER

The image shows the MIT App Inventor web interface. At the top, the MIT App Inventor logo is on the left, and navigation menus for 'Projects', 'Connect', 'Build', and 'Help' are on the right. Below the logo, the project name 'CallPhone' is displayed, along with a dropdown menu for 'Screen1' and an 'Add Screen ...' button. A context menu is open over the 'Build' menu, showing two options: 'App (provide QR code for .apk)' and 'App (save .apk to my computer)'. The 'User Interface' palette on the left lists various components like Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. The 'Viewer' window on the right shows a preview of the app's screen, titled 'Screen1', with a status bar at the top showing the time 9:48 and signal strength. The screen contains a text input field, two buttons labeled 'Call' and 'Share...', and a large image of a globe.

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 לבמייל 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

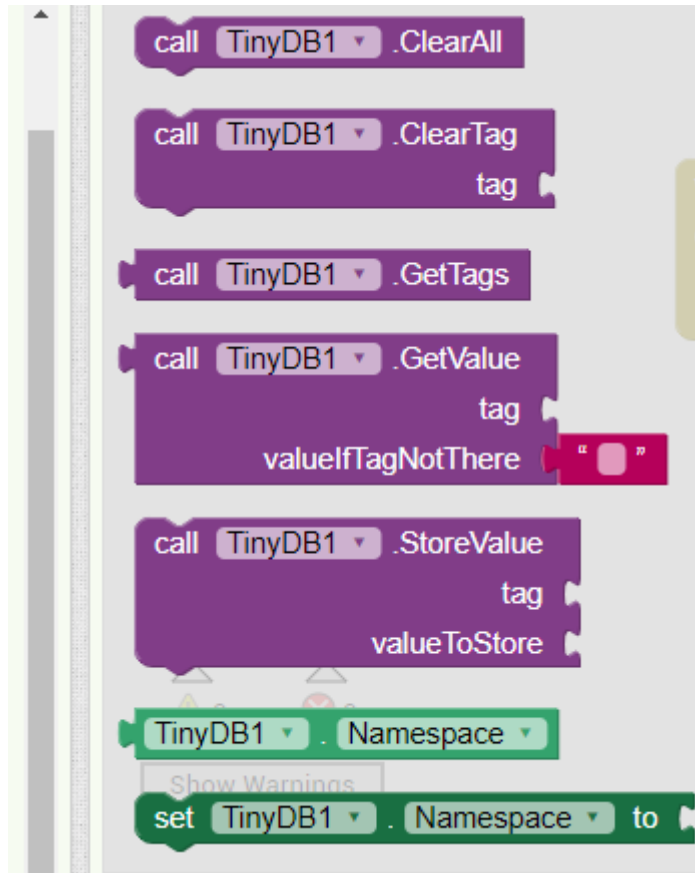
- יש לבחור את האופציה TINYDB
- יש לגרור את TINY DB על גבי המסך

Storage		
	File	
	FusiontablesControl	
	TinyDB	
	TinyWebDB	

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



```
call TinyDB1 .ClearAll
call TinyDB1 .ClearTag
tag
call TinyDB1 .GetTags
call TinyDB1 .GetValue
tag
valueIfTagNotThere " "
call TinyDB1 .StoreValue
tag
valueToStore
TinyDB1 . Namespace
Show Warnings
set TinyDB1 . Namespace to
```

- 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 get global notes  
item EntryTextBox.Text
```

```
set NotesLabel.Text to call listToText list  
get global notes
```

```
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
```

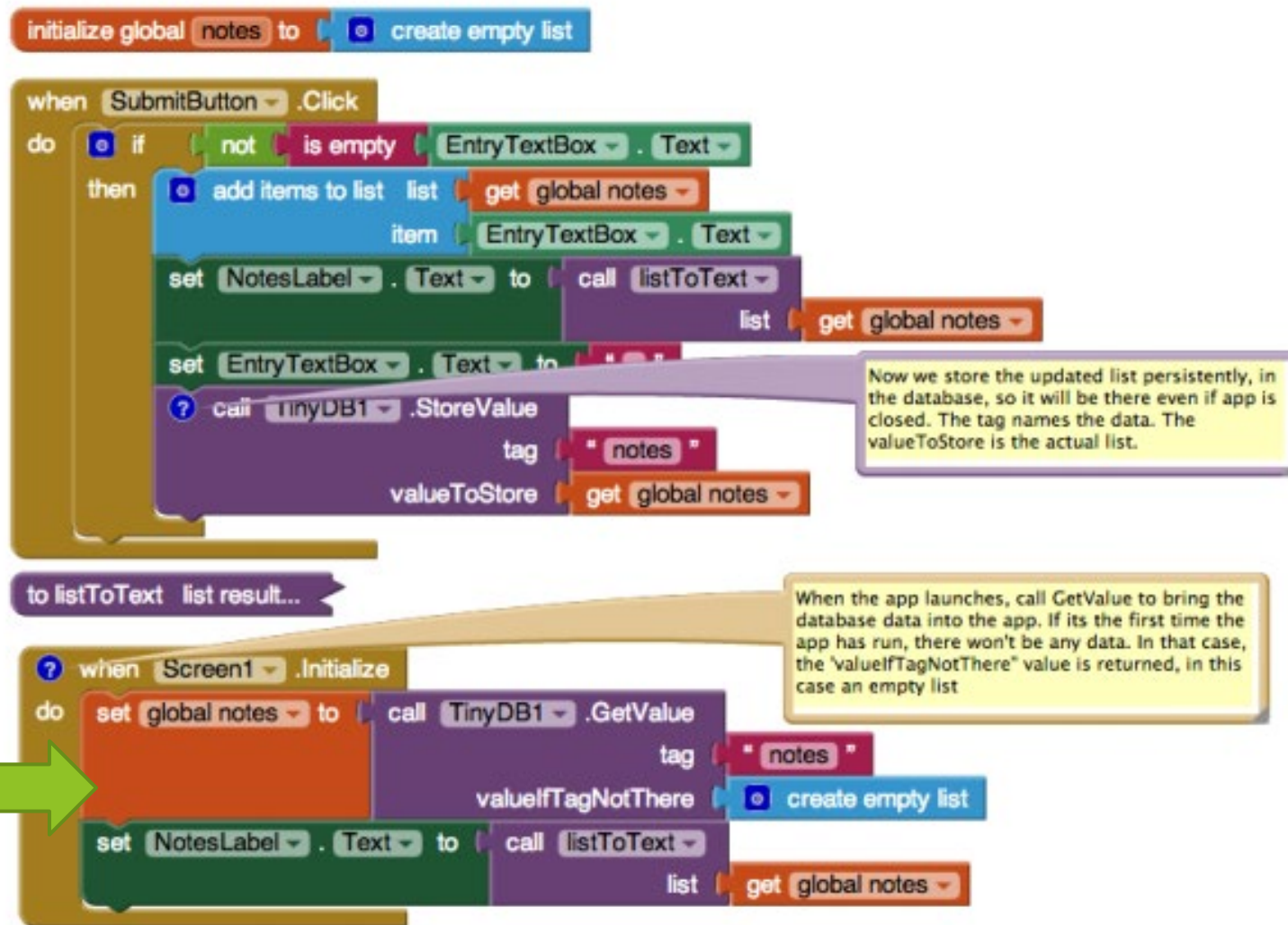
```
set NotesLabel.Text to call listToText list  
get global notes
```

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 הודעת 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

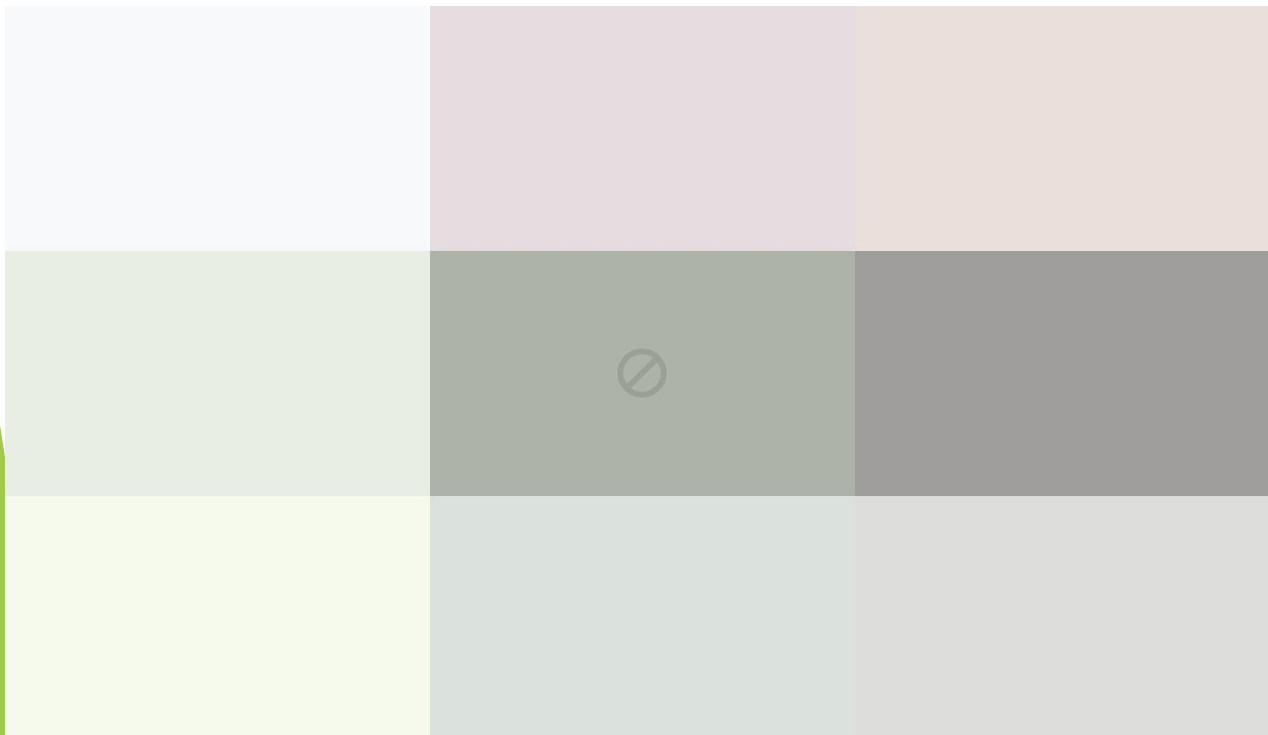
1. מה זה APPSGEYSER?

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



APPS GEYSER

- ▶ בעבודה עם APP GEYSER אין צורך בשום רקע בתכנות - אפליקציות באתר כבר מוכנות
- ▶ באתר יש מאגר גדול מאוד של אפליקציות
- ▶ כל מה שמתמש 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

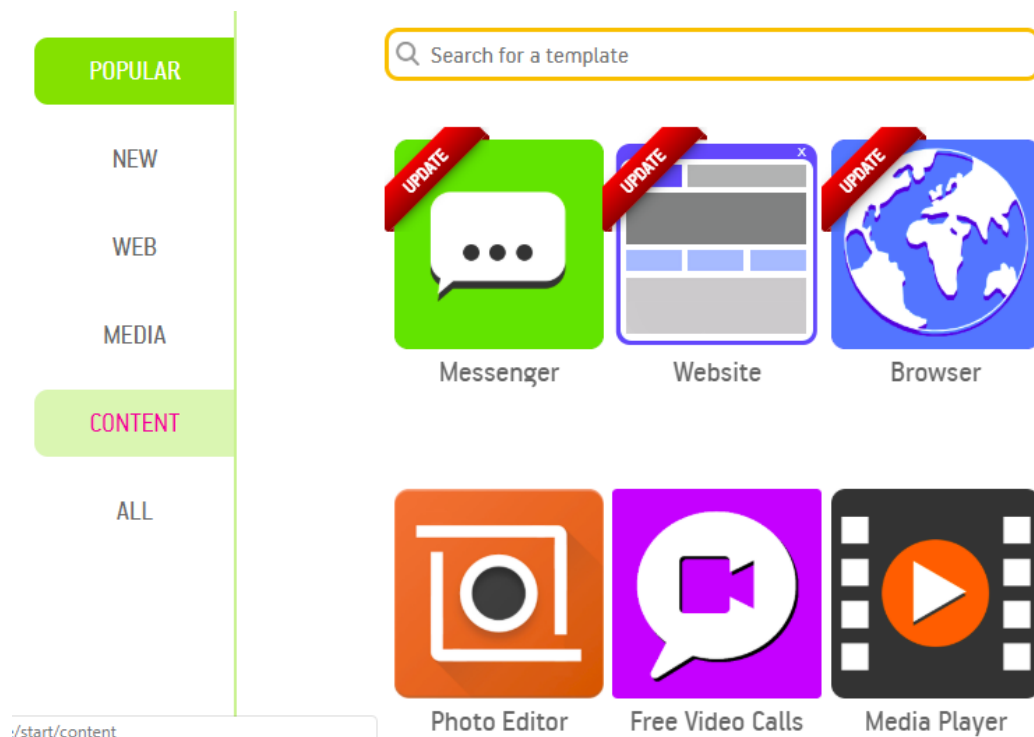
לאחר ההזדהות הראשונה, יפתח לכם החלון הראשי של APPSGEYSER ►

The screenshot shows the Appsgyser dashboard interface. On the left is a green sidebar with navigation options: DASHBOARD, PUBLISH, MONETIZE, STATISTICS, EDIT, BACKUPS, PUSH, and PREMIUM. The main content area features a large green banner with the text "You have no apps" and "Create your first app now!". Below this banner is a yellow "CREATE NOW" button. A green callout box with an arrow points to the "CREATE NOW" button, containing the text: "ההודעה מסבירה שאין לנו עדיין אף אפליקציה ויש להתחיל לפתח אותה". In the top right corner, there are buttons for "DOWNLOAD APK", "LOGOUT", and "CREATE". A notification window from "www.appsgyser.com" is open, showing "Show notifications" and "Allow/Block" options.

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

The screenshot displays the APPSGEYSER app store interface. On the left, there is a vertical navigation menu with the following categories: POPULAR (highlighted in green), NEW, WEB, MEDIA, CONTENT (highlighted in light green), and ALL. The main content area features a search bar at the top with the placeholder text "Search for a template". Below the search bar, there are six app templates arranged in two rows of three. Each template has a red "UPDATE" ribbon in the top-left corner. The first row includes: Messenger (green icon with a speech bubble), Website (blue icon with a browser window), and Browser (blue icon with a globe). The second row includes: Photo Editor (orange icon with a camera), Free Video Calls (purple icon with a video call camera), and Media Player (black icon with a play button on a film strip). The URL "/start/content" is visible at the bottom left of the interface.

/start/content

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.

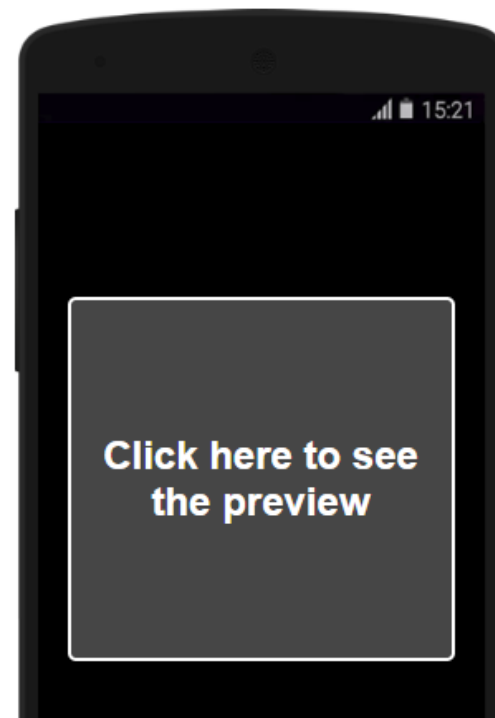
UI assets are from: <http://www.fleamedia.com/category/free-resource>

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

Fields marked with * symbol are **required**

APP SETTINGS

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



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](#)

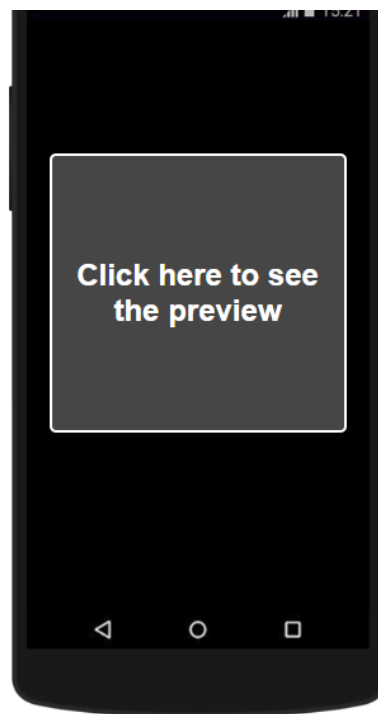
Add a easy enemy. Killed with 1 shot:

Name: *

Image: *



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](#)

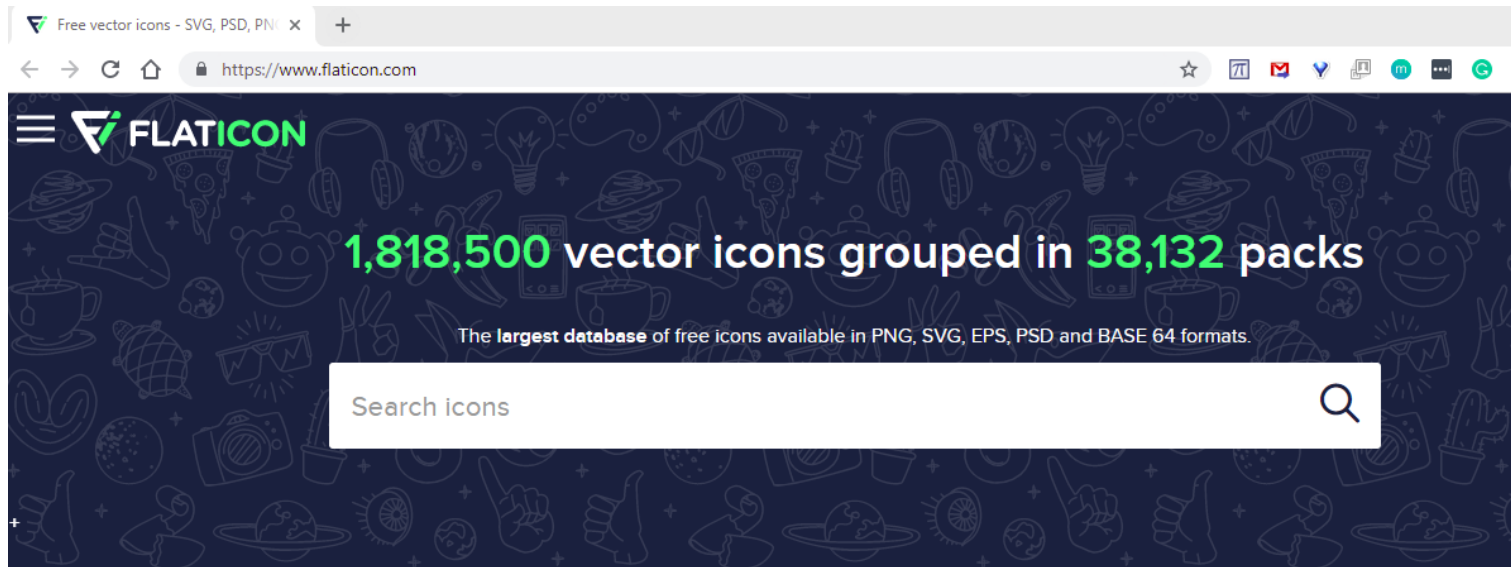


פיתוח אפליקציה ראשונה

במידה וחסר לכם תמונות

ניתן להיכנס לאתר תמונות ולמצוא תמונות נוספות

FLATICON = זה האתר המומלץ ב-APPGEYSER



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

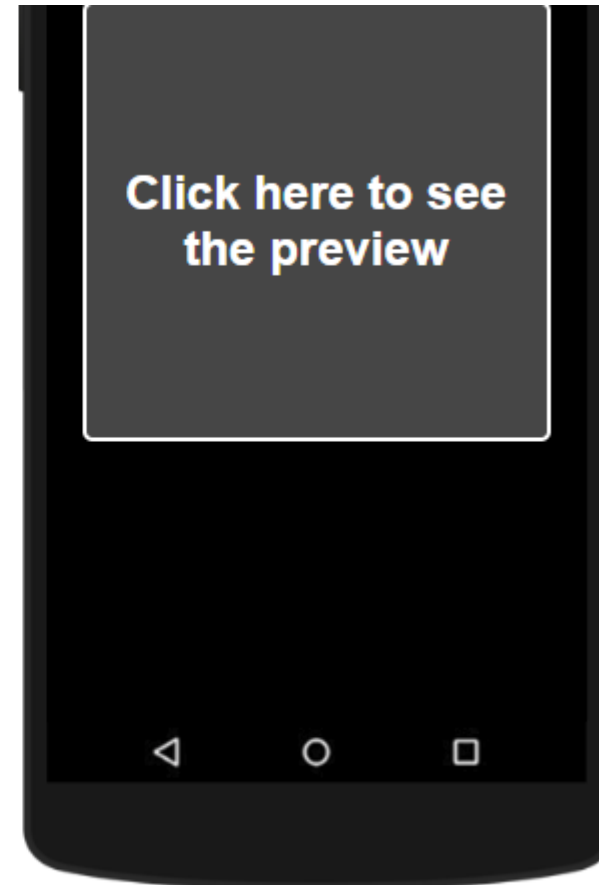
APP NAME

DESCRIPTION

ICON

CREATE

Fine tune your app after clicking Create App



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.

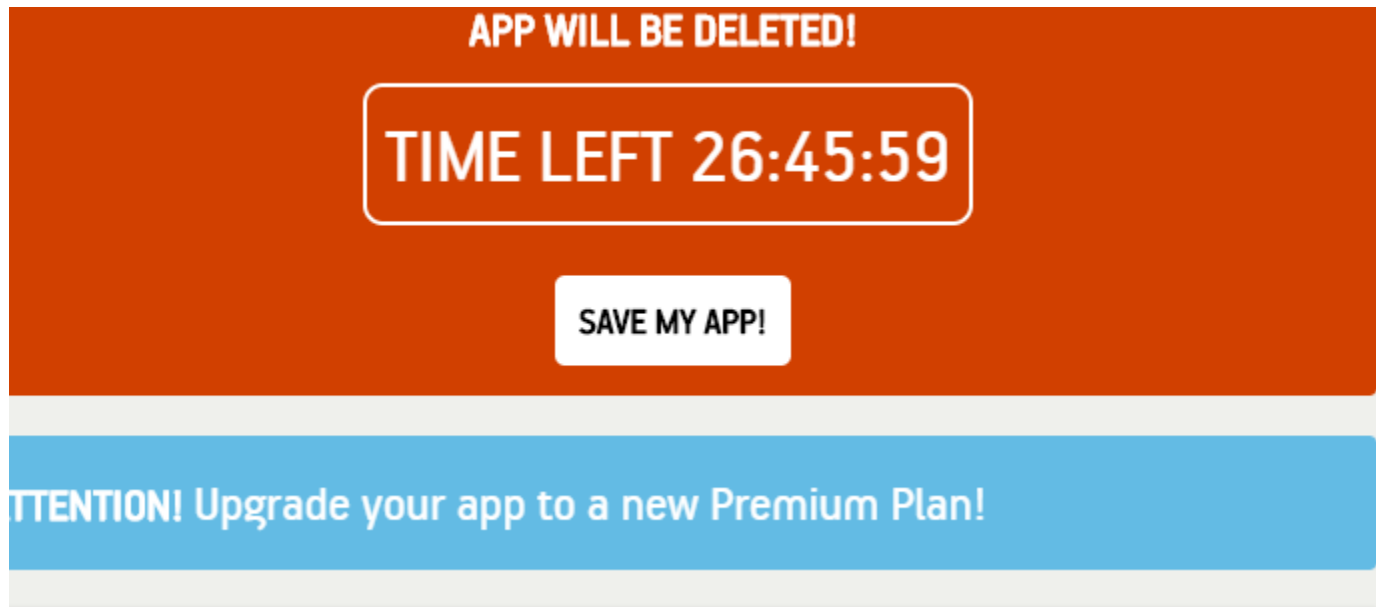
Phone *

Name

SUBMIT!

שמירת אפליקציה הראשונה

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



The image shows a screenshot of a mobile application warning dialog. The top section is a red banner with the text "APP WILL BE DELETED!". Below this, a white rounded rectangle contains a timer showing "TIME LEFT 26:45:59". Underneath the timer is a white button with the text "SAVE MY APP!". Below the red banner is a blue banner with the text "ATTENTION! Upgrade your app to a new Premium Plan!".

הורדת קובץ ה־APK

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



Select your app:



BADMAN11
app id: BADMAN11_9111876

DOWNLOAD APK

הורדת קובץ ה־APK

DASHBOARD

NEW

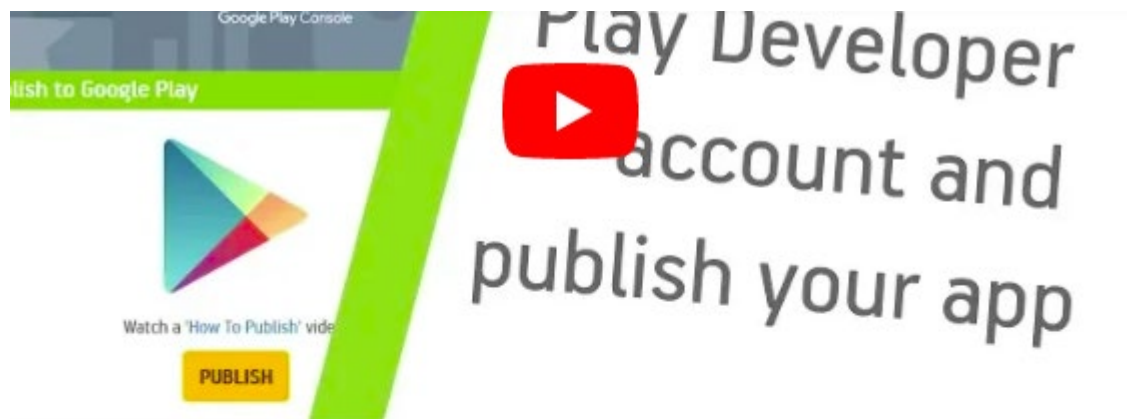
REMOVE ADS

PUBLISH APP ON GOOGLE PLAY TO SAVE IT!

TIME LEFT 26:38:48

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 ל־GOOGLE DRIVE ונשלח לעצמינו שימו לב!!! לא ניתן לשלוח את הקובץ דרך המייל מכיון 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

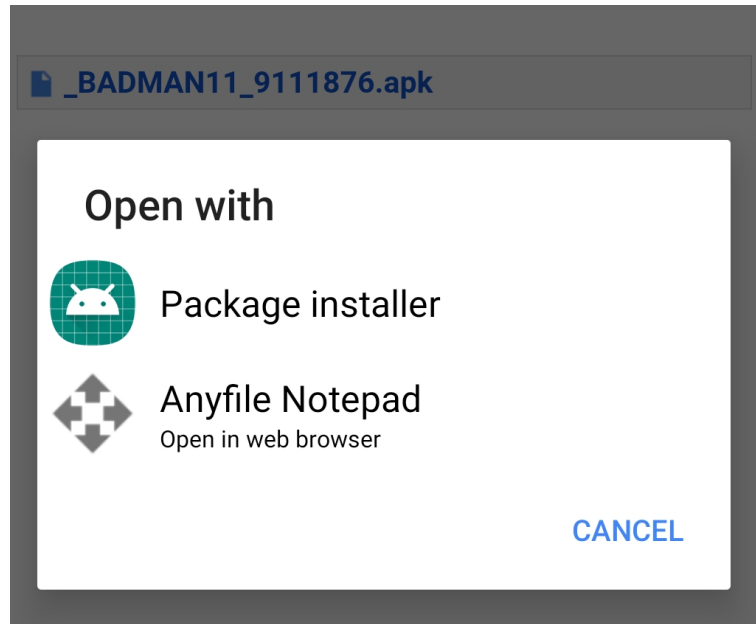
Insert as

 Drive link

 Attachment

התקנת האפליקציה

יש להוריד את הקובץ מ-GOOGLE DRIVE או מהמייל
יש ללחוץ על הקובץ ולבחור באפשרות התקנה - PACKAGE INSTALLER



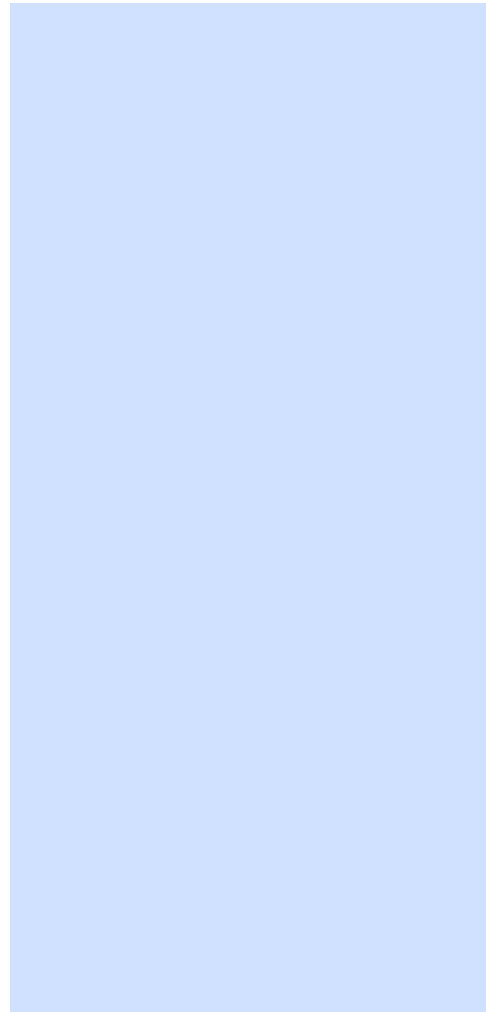
התקנת האפליקציה

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



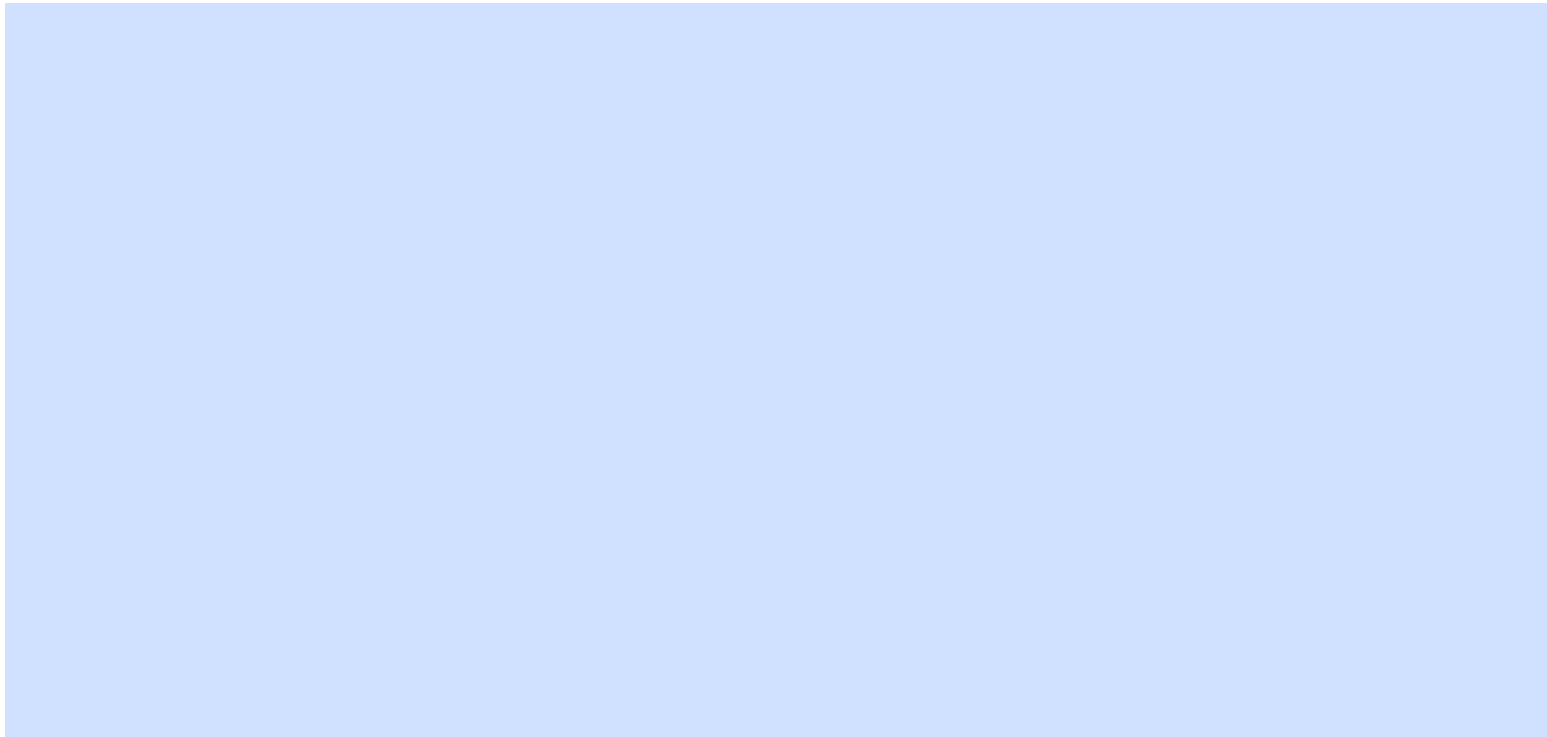
התקנת האפליקציה

גם התמונות וגם הטקסט משולבים בתוך האפליקציה



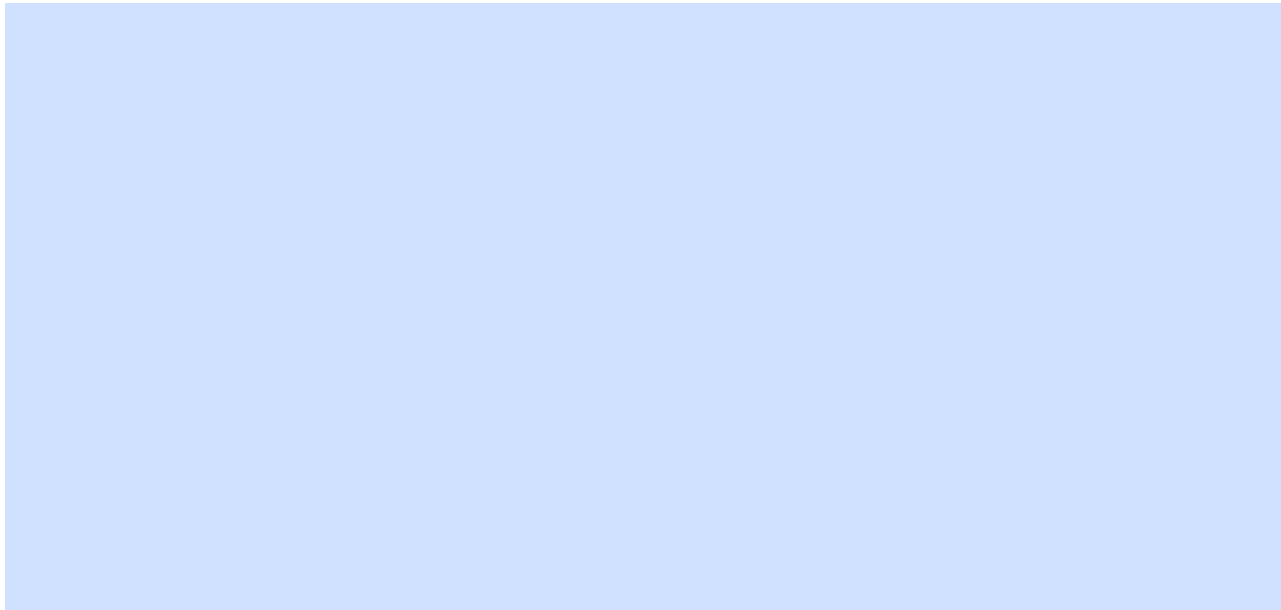
התקנת האפליקציה

גם ההסברים למשחק - מתוך הטקסט שכתבתנו עבורה



התקנת האפליקציה

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



תרגיל מספר

- יש למצוא ולהפעיל אפליקציה TIC TAC TOE באמצעות APP GEYSER
- יש למצוא ולהפעיל משחק נוסף - לפי בחירתכם