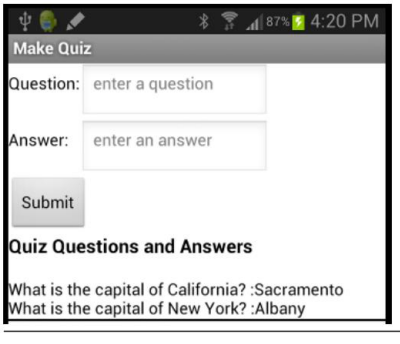
ziuQekaT-ıMakeQuiz

איור .10-1

אתה יכול להתאים אישית את אפליקציית חידון הנשיאים בפרק 8כדי לבנות כל חידון, אבל רק המתכנת יכול לשנות את השאלות והתשובות. אין דרך להורים, מורים או משתמשי אפליקציה אחרים ליצור חידונים משלהם או לשנות את שאלות החידון (אלא אם גם הם רוצים ללמוד כיצד להשתמש ב-ppA .(Inventor!

בפרק זה, תבנה אפליקציית MakeQuizהמאפשרת "מורה" יוצר חידונים באמצעות טופס קלט. השאלות והתשובות של החידון יאוחסנו במסד נתונים אינטרנטי כך ש"סטודנטים" יוכלו לגשת לאפליקציית TakeQuizנפרדת ולגשת למבחן. תוך כדי בניית שתי האפליקציות הללו, תעשה עוד קפיצת מדרגה קונספטואלית משמעותית: למד כיצד ליצור אפליקציות עם נתונים שנוצרו על ידי משתמשים המשותפים בין אפליקציות ו

משתמשים.



איור .10-2אפליקציית MakeQuiz בפעולה

הורים יכולים ליצור אפליקציות טריוויה מהנות עבור ילדיהם במהלך טיול ארוך, מורים בבית ספר תיכון יכולים לבנות חידוני "מתמטיקה מפוצץ" ותלמידי מכללה יכולים לבנות חידונים כדי לעזור לקבוצות הלימוד שלהם להתכונן לסיום. פרק זה מתבסס על חידון הנשיאים בפרק ,8כך שאם לא השלמת את האפליקציה הזו, עליך לעשות זאת לפני שתמשיך כאן.

תעצב שתי אפליקציות, MakeQuizעבור המורה (ראה איור 10-1ו-ziuQekaT עבור התלמיד, אשר יופיעו בדומה ל-
חידון נשיאים.

להלן ההתנהגויות שתקודד עבור האפליקציה הראשונה, :MakeQuiz

- המשתמש מקליד שאלות ותשובות בטופס קלט.
- צמדי השאלה-תשובה מוצגים.
- השאלות והתשובות של החידון מאוחסנות במסד נתונים אינטרנטי.

האפליקציה השנייה שתיצור, ,TakeQuizתעבוד בדומה לאפליקציית Presidents Quizשכבר בנית. למעשה, תשתמש באפליקציית חידון הנשיאים כנקודת התחלה.
TakeQuizיהיה שונה בכך שהשאלות שנשאלו יהיו אלו שהוכנסו למסד הנתונים באמצעות MakeQuiz.

מה תלמד

חידון הנשיאים היה דוגמה לאפליקציה עם נתונים סטטיים: לא משנה כמה פעמים תתמודדו עם החידון, השאלות תמיד זהות כי הן מקודדות קשיחות באפליקציה; כלומר, השאלות והתשובות הן חלק מהבלוקים. אפליקציות חדשות, בלוגים ואפליקציות רשתות חברתיות כמו פייסבוק וטוויטר פועלות עם נתונים דינמיים, כלומר הנתונים יכולים להשתנות לאורך זמן. לעתים קרובות, מידע דינמי זה נוצר על ידי המשתמש -האפליקציה מאפשרת למשתמשים להזין, לשנות ולשתף מידע. עם MakeQuizו-ziuQekaT, תלמד כיצד לבנות אפליקציה שמטפלת בנתונים משותפים שנוצרו על ידי משתמשים.

אם השלמת את אפליקציית Xylophone(פרק ,9כבר הוצגת בפניך לרשימות דינמיות; באפליקציה זו, התווים המוזיקליים שהמשתמש מנגן מוקלטים ברשימות. אפליקציות עם נתונים כאלה שנוצרו על ידי משתמשים מורכבות יותר, והבלוקים מופשטים יותר מכיוון שהם לא מסתמכים על נתונים סטטיים מוגדרים מראש. אתה מגדיר משתני רשימה, אבל אתה מגדיר אותם ללא פריטים ספציפיים. בזמן שאתה מתכנת את האפליקציה שלך, אתה צריך לדמיין את הרשימות מאוכלסות בנתונים שמסופקים על ידי משתמש הקצה.
מדריך זה מכסה את המושגים הבאים של ממציא האפליקציות:

•טפסי קלט לאפשר למשתמש להזין מידע.

•שימוש ברשימה באינדקס יחד עם עבור כל אחד מהם כדי להציג פריטים ממספר רשימות.

•נתוני רשימה מתמשכת MakeQuiz -ישמור את השאלות והתשובות של החידון במסד נתונים אינטרנטי, ו-ziuQekaT יטען אותן מאותו מסד נתונים.

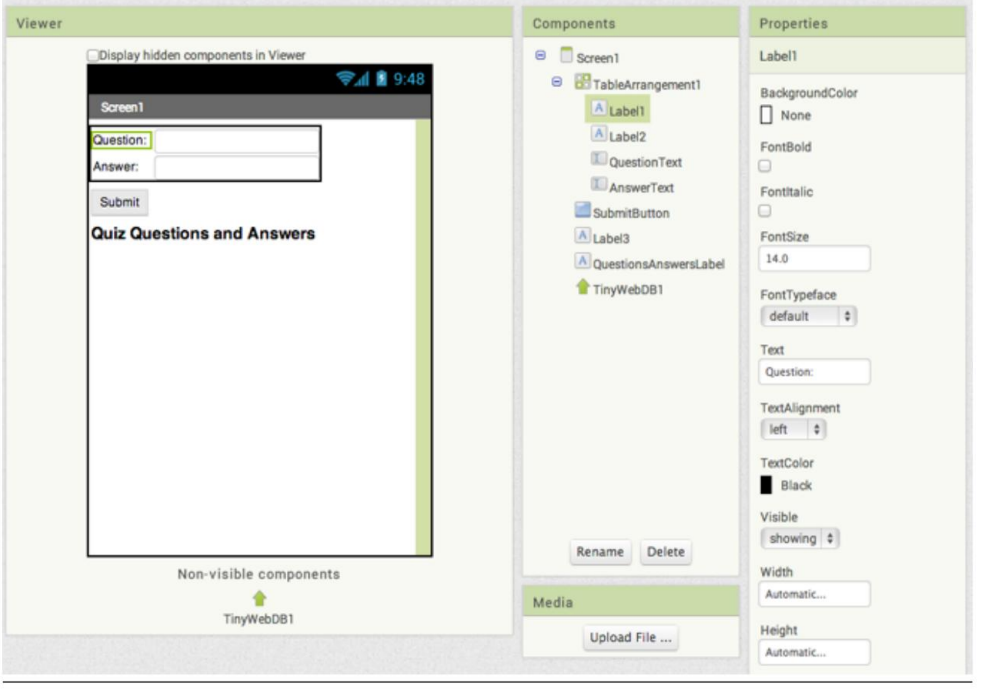
•שיתוף נתונים -אתה תשמור את הנתונים במסד נתונים אינטרנטי באמצעות רכיב TinyWebDB(במקום רכיב TinyDBששימש בפרקים הקודמים).

מתחילים

התחבר לאתר App Inventorוהתחל פרויקט חדש. תן לזה שם "MakeQuiz"והגדר את כותרת המסך ."Make Quiz"לחבר את האפליקציה שלך למכשיר או לאמולטור שלך לבדיקה חיה.

עיצוב הרכיבים

השתמש Component Designer-בכדי ליצור את הממשק עבור .MakeQuizכשאתה מסיים, זה צריך להיראות משהו כמו איור 10-2(יש גם הוראות מפורטות יותר לאחר מכן את תמונת המצב).



איור 10-3. MakeQuiz ב-Component Designer

אתה יכול לבנות את ממשק המשתמש המוצג באיור 10-2על ידי גרירת ה-רכיבים המפורטים בטבלה .10-1גרור כל רכיב מהלוח לתוך ה-מציג ושם אותו כפי שצוין בטבלה. שים לב שאתה יכול להשאיר את תווית הכותרת שמות (Label1 - Label4)כברירת המחדל שלהם (לא תשתמש בהם בעורך הבלוקים בכל מקרה).

טבלה .10-1כל הרכיבים לאפליקציית MakeQuiz

מַטָרָה	קבוצת צבעים איך תקרא לזה		סוג רכיב
עיצוב הטופס, כולל השאלה ו תשובה.	פחיסדהשולחן	שולחן	
ההנחיה "שאלה:".	תווית ממשק משתמש 1		תווית
המשתמש מזין שאלות כאן.	ממשק משתמש QuestionText		תיבת טקסט

מַטָרָה	קבוצת צבעים איך תקרא לזה	סוג רכיב
ההנחיה "תשובה:".	תווית ממשק משתמש2	תווית
המשתמש מזין כאן תשובות.	ממשק משתמש AnswerText	תיבת טקסט
המשתמש לוחץ על זה כדי לשלוח צמד QA.	ממשק משתמש SubmitButton	לְחִצָן
הצג את "שאלות ותשובות חידון".	תווית ממשק משתמש 3	תווית
שאלות ממשק משתמש תשובות תווית הצגת צמדי QAשהוזנו בעבר.		תווית
אִחסוֹן אחסון אינטרנט עבור זוגות QA.	TinyWebDB1	TinyWebDB

הגדר את המאפיינים של הרכיבים בצורה הבאה:

1. הגדר את הטקסט של תווית1 ל"שאלה", את הטקסט של תווית2 ל"תשובה ", ואת טקסט של 3lebaL ל"שאלות ותשובות חידון".

2. הגדר את FontSizeשל 3lebaL ל-81 וסמן את התיבה FontBold.

3. הגדר את הרמז של QuestionTextל"הזן שאלה" ואת הרמז AnswerText-ל ל"הזן תשובה".

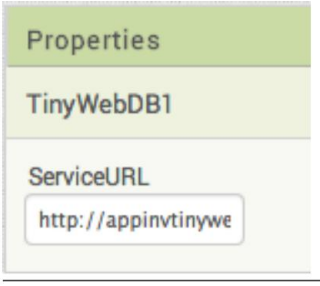
4. הגדר את הטקסט של כפתור Submitל-"timbuS".

5. הגדר את הטקסט של QuestionsAnswersLabelל"שאלות ותשובות חידון".

6. העבר את QuestionText, AnswerTextוהתוויות המשויכות אליהם לתוך סידור השולחן 1.

אם תסתכל על המאפיינים של TinyWebDB, תבחין שיש לו מאפיין ServiceURL(ראה איור 10-3). מאפיין זה מציין שירות מסד נתונים אינטרנטי, במיוחד מוגדר לעבוד עם רכיב TinyWebDB, היכן הנתונים המשותפים שלך יהיו מְאוּחסָן. כברירת מחדל, שירות האינטרנט שאליו הוא מתייחס הוא כזה שהוגדר על ידי ממציא האפליקציות של MIT צוות בכתובת http://appinvtinywebdb.appspot.com. אתה תשתמש בשירות ברירת המחדל הזה בזה הדרכה תוך כדי עבודה; עם זאת, חשוב לדעת שכל מי שמשתמש ב-Inventor ppA יאחסן מידע באותו שירות אינטרנט, וכי הנתונים שהאפליקציה שלך מציבה כולם יראו, ואולי אף ידרוס על ידי מישהו.

שירות ברירת המחדל מיועד לבדיקה בלבד. זה די קל (וגם בחינם!) להגדיר את שלך בעל שירות כזה, שתרצה לעשות אם תבנה אפליקציה שתיפרס עם משתמשים אמיתיים. לעת עתה, המשך והשלם את המדריך הזה, אבל כשתגיע מוכן ההוראות להגדרת שירות האינטרנט שלך נמצאות ב-"BDbeWyniT and ממשקי APIתואמי "TinyWebDB בעמוד 368.



איור .10-4עם TinyWebDB.ServiceURL, אתה יכול לציין את כתובת האתר של מסד נתונים אינטרנט שהגדרת

הוספת התנהגויות לרכיבים

כמו באפליקציית חידון הנשיאים, תחילה תגדיר כמה משתנים גלובליים עבור QuestionList ו-AnswerList, אבל הפעם לא תספק שאלות קבועות ו תשובות.

יצירת רשימות שאלות ותשובות ריקות

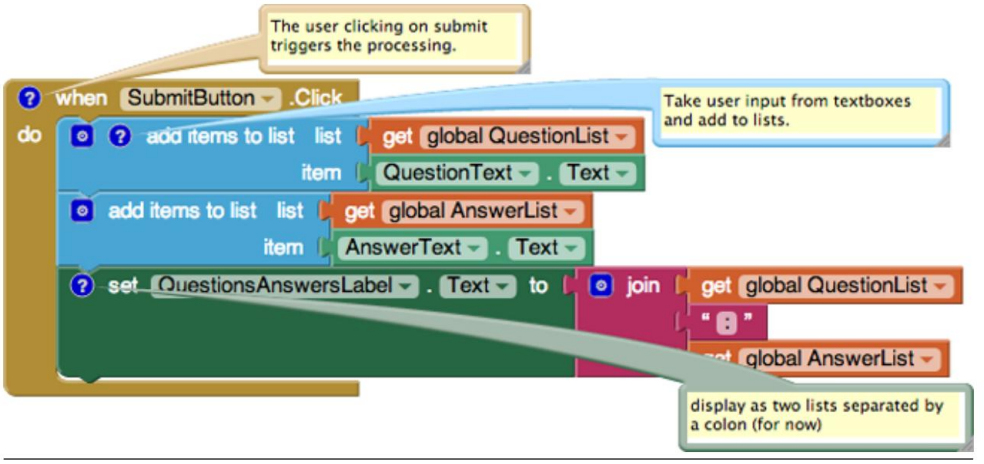
הבלוקים עבור הרשימות צריכים להיראות כפי שמוצג באיור .10-4

איור .10-5הרשימות של MakeQuiz מתחילות ריקות

הרשימות מוגדרות באמצעות בלוק צור רשימה ריקה , במקום בלוק הפוך לרשימה . הסיבה לכך היא שעם האפליקציות MakeQuiz-I ,ziuQekaT, כל הנתונים ייצרו על ידי משתמש האפליקציה (זהו נתונים דינמיים שנוצרו על ידי המשתמש).

הקלטת כניסות המשתמש

ההתנהגות הראשונה שתבנה היא לטיפול בקלט של המשתמש. באופן ספציפי, כאשר המשתמש מזין שאלה ותשובה ולוחץ על שלח, תשתמש בהוספת פריטים לבלוקים ברשימה כדי לעדכן את רשימת השאלות והתשובות . הבלוקים צריכים להופיע כמתואר באיור .10-5



איור 10-6. הוספת ערכים חדשים לרשימות

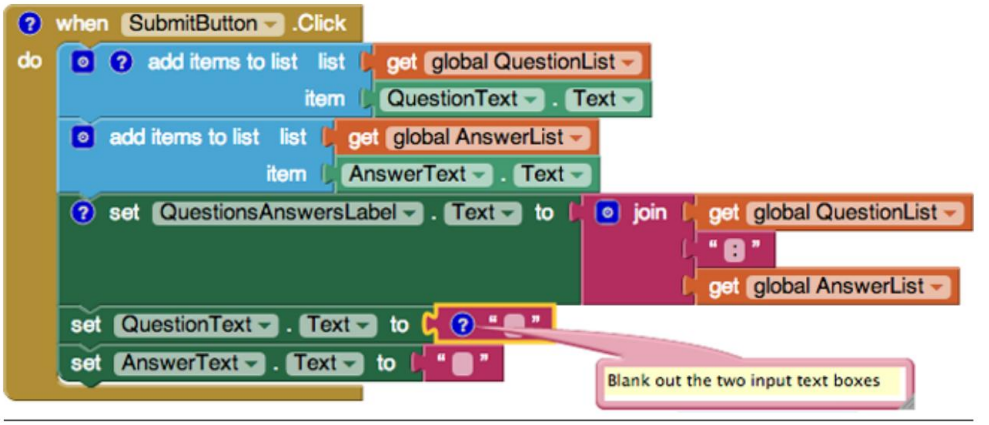
איך הבלוקים עובדים

גוש הסף פריטים לרשימה מוסיף כל פריט לסוף רשימה. כפי שמוצג באיור 10-5, האפליקציה לוקחת את הטקסט שהמשתמש הזין בתיבות הטקסט QuestionTextו-AnswerTextומצרפת כל אחת מהן לרשימה המתאימה.

הוספת הפריטים לרשימה בלוקים מעדכנים את המשתנים QuestionListו- , AnswerList אך שינויים אלה עדיין לא מוצגים למשתמש. השורה השלישית של בלוקים מציגה את הרשימות הללו על ידי שרשורן (הצטרפותן) עם נקודתיים מוכנס ביניהן. כברירת מחדל, App Inventor מציג רשימות עם סוגריים מסביב ורווחים בין פריטים: לדוגמה, "(פריט1 פריט2 פריט3)." כמובן שזו לא הדרך האידיאלית להציג את הרשימות, אבל היא תאפשר לכם לבדוק את התנהגות האפליקציה לעת עתה. מאוחר יותר, תיצור שיטה מתוחכמת יותר להצגת הרשימות המציגות כל צמד תשובות לשאלה בשורה נפרדת.

מחיקת השאלה והתשובה

נזכיר מאפליקציית Presidents Quizשכשעברתם לשאלה הבאה ברשימה, הייתם צריכים לבטל את תשובת המשתמש מהשאלה הקודמת. באפליקציה זו, כאשר משתמש שולח צמד שאלות ותשובות, תרצה לנקות את תיבות הטקסט QuestionTextו- AnswerTextכך שהם יהיו מוכנים לערך חדש במקום להציג את הקודמת. הבלוקים צריכים להופיע כפי שמוצגים באיור 10-6.



איור .7-10ביטול תיבות הטקסט של השאלה והתשובות לאחר ההגשה

בדוק את האפליקציה שלך בדוק את ההתנהגות על ידי הזנת כמה זוגות שאלות ותשובות. כאשר אתה מוסיף אותם, האם הם מופיעים מתחת לטופס בתווית ?QuestionsAnswersLabel

איך הבלוקים עובדים

כאשר המשתמש שולח שאלה ותשובה חדשים, הם מתווספים לרשימות המתאימות ומוצגים. בשלב זה, הטקסט ב- QuestionTextו- AnswerTextנטוש בלוקי טקסט ריקים.

הצגת צמדי שאלה-תשובה במספר קווים

באפליקציה שבניתם עד כה, רשימות השאלות והתשובות מוצגות בנפרד ובשימוש בפורמט תצוגת הרשימה המוגדרת כברירת מחדל עבור App Inventor. אז אם היית עושה חידון על בירות המדינה והזנת שני זוגות של שאלות ותשובות, זה עשוי מופיע כ:

(מהי בירת קליפורניה? מהי בירת ניו יורק?: סקרמנטו אלבני)

זה כמובן לא ממשק משתמש אידיאלי עבור מעצב החידון. תצוגה טובה יותר תציג כל שאלה יחד עם התשובה המתאימה לה, עם צמד תשובות שאלה בשורה, כך:

מהי בירת קליפורניה?: סקרמנטו מהי בירת ניו יורק?: אלבני הטכניקה להצגת רשימה בודדת עם כל פריט בשורה נפרדת מתוארת בפרק - 20כדאי שתקרא את הפרק הזה לפני שתמשיך.

המשימה כאן קצת יותר מסובכת כי אתה מתמודד עם שתי רשימות. בגלל המורכבות שלו, תכניס את הבלוקים להצגת הנתונים בהליך

בשם displayQAsוקראו לנוהל הזה מהמטפל באירועים . SubmitButton.Click

כדי להציג צמדי שאלות ותשובות בשורות נפרדות, יהיה עליך לבצע את הפעולות הבאות:

•השתמש ב-a עבור כל בלוק כדי לחזור על כל שאלה ברשימת השאלות.

•השתמש ב- answerIndexמשתנה כדי שתוכל לתפוס כל תשובה בזמן שאתה חוזר על השאלות.

•השתמש ב- joinכדי לבנות אובייקט טקסט עם כל צמד שאלה ותשובות, ו-א תו קו חדש (\n)המפריד בין כל זוג.

הבלוקים צריכים להופיע כמתואר באיור .10-7

איור .10-8נוהל displayQAs

איך הבלוקים עובדים

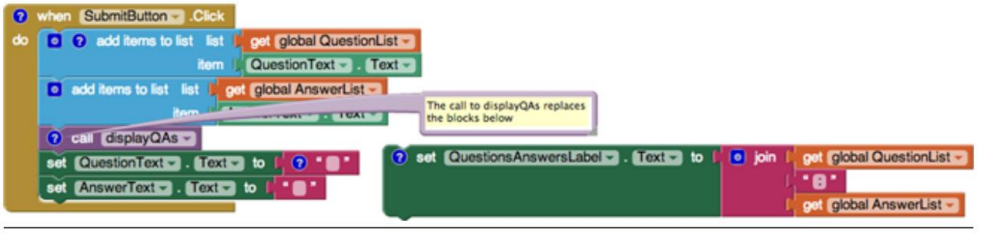
הליך displayQAsמקיף את כל הבלוקים להצגת הנתונים.
על ידי שימוש בהליך, לא תצטרך להעתיק את הבלוקים הדרושים להצגת הרשימה
יותר מפעם אחת באפליקציה -אתה יכול פשוט להתקשר ל-sAQyalpsid כאשר אתה צריך להציג את הרשימות.

עבור כל אחד בלבד מאפשר לך לחזור על רשימה בודדת. במקרה הזה, יש שתי רשימות, QuestionListו- .AnswerListהערך עבור כל אחד משמש כדי לחזור על רשימת השאלות, אבל אתה צריך לבחור תשובה, כמו גם, כשאתה ממשיך בשאלות. כדי להשיג זאת, אתה משתמש במשתנה אינדקס, כפי שנעשה עם ה- currentQuestionIndexבמדריך חידון הנשיאים בפרק .8במקרה זה, משתנה האינדקס, answerIndexמשמש כדי לעקוב אחר המיקום ב- AnswerListכאשר עבור כל אחד מהם עובר רשימת השאלות.

answerIndexמוגדר ל-1 לפני ההתחלה של עבור כל אחד . בתוך עבור כל אחד, answerIndexמשמש לבחירת התשובה הנוכחית מרשימת התשובות, ולאחר מכן היא מוגדלת. בכל איטרציה של עבור כל אחד, השאלה והתשובה הנוכחיות משורשרות לסוף המאפיין , QuestionsAnswersLabel.Textעם נקודתיים ביניהן.

התקשרות לנוהל התצוגה

כעת יש לך נוהל להצגת צמדי שאלות ותשובות, אבל זה לא יעזור אלא אם תתקשר אליו כשתצטרך. שנה את המטפל באירועים SubmitButton.Clickעל ידי קריאת displayQAs במקום הצגת הרשימות, כפי שנעשה קודם לכן. הבלוקים המעודכנים צריכים להופיע כפי שמוצג באיור .10-8



איור .10-9קורא לנוהל displayQAsכדי להחליף את הבלוקים המוצגים מימין

בדוק את האפליקציה שלך בדוק את ההתנהגות על ידי הזנת כמה זוגות שאלות ותשובות. האם בזמן הוספתם הם מופיעים בשורות נפרדות בתווית ?QuestionsAnswersLabel

שמירת ה-SAQ בהתמדה באינטרנט

עד כה, יצרת אפליקציה שממקמת את השאלות והתשובות שהוזנו ברשימה. אבל מה קורה אם יוצר החידון סוגר את האפליקציה? אם השלמת את האפליקציה ללא הודעות טקסט בזמן נהיגה (פרק 4)או את האנדרואיד, איפה המכונית שלי? אפליקציה (פרק ,(7אתה יודע שאם לא תשמור את הנתונים במסד נתונים, הם לא יהיו שם כשהמשתמש יוצא ומפעיל מחדש את האפליקציה. אחסון הנתונים באופן מתמיד יאפשר ליוצר החידון להציג או לערוך את העדכון האחרון של החידון בכל פעם שהאפליקציה מופעלת. אחסון מתמשך נחוץ גם מכיוון שאפליקציית TakeQuizזקוקה גם לגישה לנתונים.

אתה כבר מכיר את השימוש ברכיב TinyDBלאחסון ואחזור נתונים במסד נתונים. אבל במקרה זה, במקום זאת, תשתמש ברכיב TinyWebDB . TinyWebDBבעוד TinyDBמאחסנת מידע ישירות בטלפון, TinyWebDBמאחסנת נתונים במסדי נתונים השוכנים באינטרנט.

מה לגבי עיצוב האפליקציה שלך יזכה להשתמש במסד נתונים מקוון במקום אחד המאוחסן בטלפון של אדם? הבעיה העיקרית כאן היא שאתה בונה שתי אפליקציות כאלה

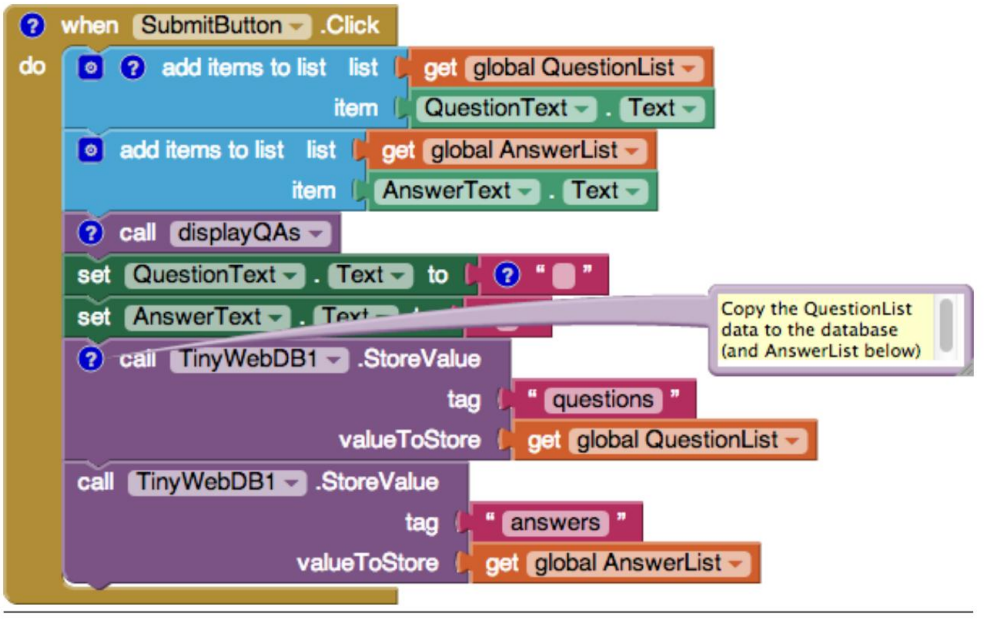
שניהם צריכים גישה לאותם נתונים -אם יצרנית החידון מאחסנת את השאלות והתשובות בטלפון שלה, למתמודדי החידון לא תהיה שום דרך להגיע לנתונים עבור החידון שלהם! מכיוון ש- TinyWebDBמאחסנת נתונים באינטרנט, הנבחן יכול לגשת לשאלות ולתשובות של החידון במכשיר שונה מזה של יצרן החידון. (אחסון נתונים מקוון מכונה לעתים קרובות ענן.)
להלן הסכימה הכללית להפיכת נתוני רשימה (כגון השאלות והתשובות עבור האפליקציה שלנו) לעמידות:

- אחסן רשימה במסד הנתונים בכל פעם שמתווסף לו פריט חדש.
- כאשר האפליקציה מופעלת, טען את הרשימה ממסד הנתונים למשתנה.

התחל על ידי אחסון השאלות והתשובות רשימת במסד הנתונים בכל פעם שהמשתמש מזין זוג חדש.

איך הבלוקים עובדים

ה- TinyWebDB1.StoreValueחוסם נתוני אחסון במסד נתונים אינטרנטי. ל- StoreValue יש שני ארגומנטים: התג המזהה את הנתונים, והערך שהוא הנתונים בפועל שברצונך לאחסן. איור 10-9מראה ש- QuestionListמאוחסן עם תג של "שאלות", ואילו ה- AnswerList מאוחסן עם תג של "תשובות".



איור .10-10אחסון השאלות והתשובות במאגר

עבור האפליקציה שלך, עליך להשתמש בתגים ייחודיים יותר מ"שאלות" ו"תשובות" (למשל, "DavesQuestions"ו-"srewsnAsevaD"). זה חשוב כי, ב

לפחות בהתחלה, אתה משתמש בשירות מסד הנתונים האינטרנטי המוגדר כברירת מחדל עבור App Inventor, מה שאומר שאחרים יכולים להחליף את השאלות והתשובות שלך, כולל אנשים אחרים שעוקבים אחר המדריך הזה.

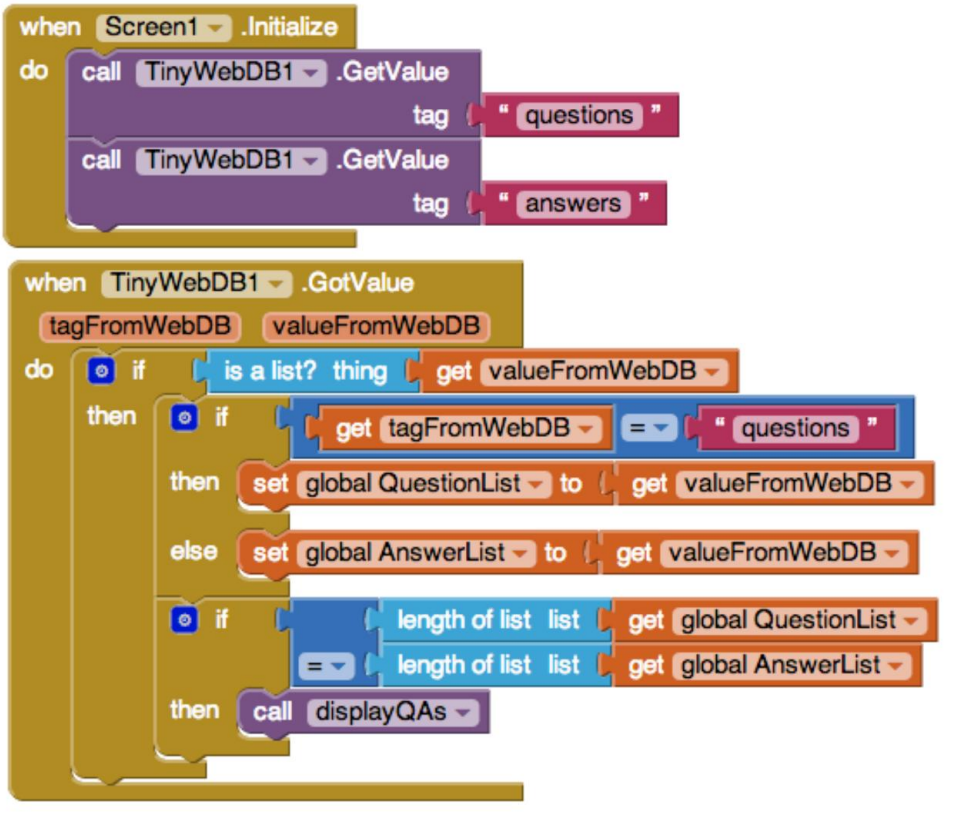
בדוק את האפליקציה שלך בדיקת חלק זה של האפליקציה שונה מבדיקות שביצעת בעבר מכיוון שהאפליקציה שלך משפיעה כעת על ישות אחרת, שירות ברירת המחדל של TinyDBWeb. הפעל את האפליקציה, הזן שאלה ותשובה ולאחר מכן פתח חלון דפדפן לשירות האינטרנט המוגדר כברירת מחדל בכתובת http://appinvtinywebdb.appspot.com . לאחר מכן לחץ על "get_value" והזן אחד מהתגים שלך (בדוגמה זו, "שאלות" או "תשובות"). אם הדברים פועלים כשורה, רשימות השאלות והתשובות שלך אמורות להופיע.

כפי שהוזכר קודם, שירות האינטרנט המוגדר כברירת מחדל משותף בין מתכנתים ואפליקציות, כך שהוא מיועד רק לבדיקה. כשתהיה מוכן לפרוס את האפליקציה שלך עם משתמשים אמיתיים, תרצה להגדיר שירות מסד נתונים פרטי משלך. למרבה המזל, לעשות זאת הוא פשוט ואינו דורש תכנות (ראה "ממשקי APIתואמים TinyWebDB ו-TinyWebDBeWyniT" בעמוד 368).

טוען נתונים מבסיס הנתונים

אחת הסיבות שאנו צריכים לאחסן את השאלות והתשובות במסד נתונים היא לאפשר לאדם שיוצר את החידון לסגור את האפליקציה ולהפעיל אותה מחדש במועד מאוחר יותר מבלי לאבד את השאלות והתשובות שהוקלדו קודם לכן. (אנחנו גם עושים את זה כדי שהמבחן יוכל לגשת לשאלות, אבל נתייחס לזה מאוחר יותר.) בואו נתכנת את הבלוקים לטעינת הרשימות חזרה לאפליקציה ממסד הנתונים האינטרנטי בכל פעם שהאפליקציה מופעלת מחדש.

כפי שכיסינו בפרקים קודמים, כדי לציין מה צריך לקרות כאשר אפליקציה מופעלת, אתה מתכנת את המטפל באירוע Screen.Initialize . במקרה זה, האפליקציה צריכה לבקש שתי רשימות ממסד הנתונים האינטרנטי של TinyWebDB - השאלות והתשובות - כך Screen1.Initialize-השיבצע שתי שיחות ל- TinyWebDB.GetValue. הבלוקים צריכים להופיע כמתואר באיור 10-10.



איור 10-11. בקשת הרשימות ממסד הנתונים עם פתיחת האפליקציה ועיבוד כאשר הרשימות מגיעות

איך הבלוקים עובדים

בלוקים של TinyWebDB.GetValue באיור 10-10 פועלים בצורה שונה מזו של TinyDB.GetValue, שמחזירה ערך באופן מיידי. TinyWebDB.GetValue מבקשת רק את הנתונים ממסד הנתונים האינטרנטי; הוא לא מקבל מיד ערך. במקום זאת, כאשר הנתונים מגיעים ממסד הנתונים האינטרנטי, מופעל אירוע TinyWebDB.GotValue . עליך גם לתכנת את אותו מטפל באירועים לעבד את הנתונים המוחזרים.

כאשר מתרחש אירוע TinyWebDB.GotValue , הנתונים המבוקשים כלולים ב-an ארגומנט בשם valueFromWebDB. התג שביקשת כלול בתג הארגומנט FromWebDB.

באפליקציה זו, מכיוון שמבוצעות שתי בקשות שונות עבור השאלות והתשובות, GotValue יופעל פעמיים. כדי להימנע מהכנסת שאלות ברשימת התשובות שלך, או להיפך, האפליקציה שלך צריכה לבדוק את התג כדי לראות איזו בקשה הגיעה ולאחר מכן להכניס את הערך שהוחזר ממסד הנתונים לרשימה המתאימה (שאלות או רשימת תשובות).

ב- ,Screen.Initializeהאפליקציה קוראת ל- TinyWebDB1.GetValueפעמיים: פעם אחת כדי לבקש את השאלות המאוחסנות, ופעם אחת כדי לבקש את ה- AnswerListהמאוחסנת. כאשר הנתונים מגיעים ממסד הנתונים של האינטרנט מכל אחת מהבקשות, אירוע TinyWebDB1.GotValue מופעל.

הארגומנט valueFromWebDBשל GotValueמחזיק את הנתונים המוחזרים מבקשת מסד הנתונים. אתה צריך את החסימה החיצונית אם במטפל האירוע כי מסד הנתונים יחזיר טקסט ריק ("")ב- valueFromWebDBאם זו הפעם הראשונה שבה נעשה שימוש באפליקציה ועדיין אין שאלות ותשובות. אם תשאל אם ה- valueFromWebDBהוא רשימה?, אתה מוודא שיש נתונים שהוחזרו בפועל.

אם אין נתונים כלשהם, תעקוף את החסימות לעיבודם.

אם נתונים מוחזרים (האם רשימה? היא נכונה), החסימות ממשיכות לבדוק איזו בקשה הגיעה. התג המזהה את הנתונים נמצא ב- :tagFromWebDBזה יהיה "שאלות" או "תשובות". אם התג הוא "שאלות", ה- valueFromWebDBמוכנס למשתנה .QuestionListאחרת (אחר), הוא ממוקם ברשימת התשובות. (אם השתמשת בתגים שאינם "שאלות" ו"תשובות", בדוק את אלה, במקום זאת.)

אתה רוצה להציג את הרשימות רק לאחר ששתיהן הגיעו GotValue)הופעל פעמיים). האם אתה יכול לחשוב איך תדע בוודאות ששתי הרשימות נטענות ממסד הנתונים? הבלוקים המוצגים משתמשים במבחן אם כדי לבדוק אם

האורך של הרשימות זהה, מכיוון שזה יכול להיות נכון רק אם שתי הרשימות הוחזרו. אם כן, הליך displayQAsהשימושי שכתבת קודם נקרא כדי להציג את הנתונים הטעונים.

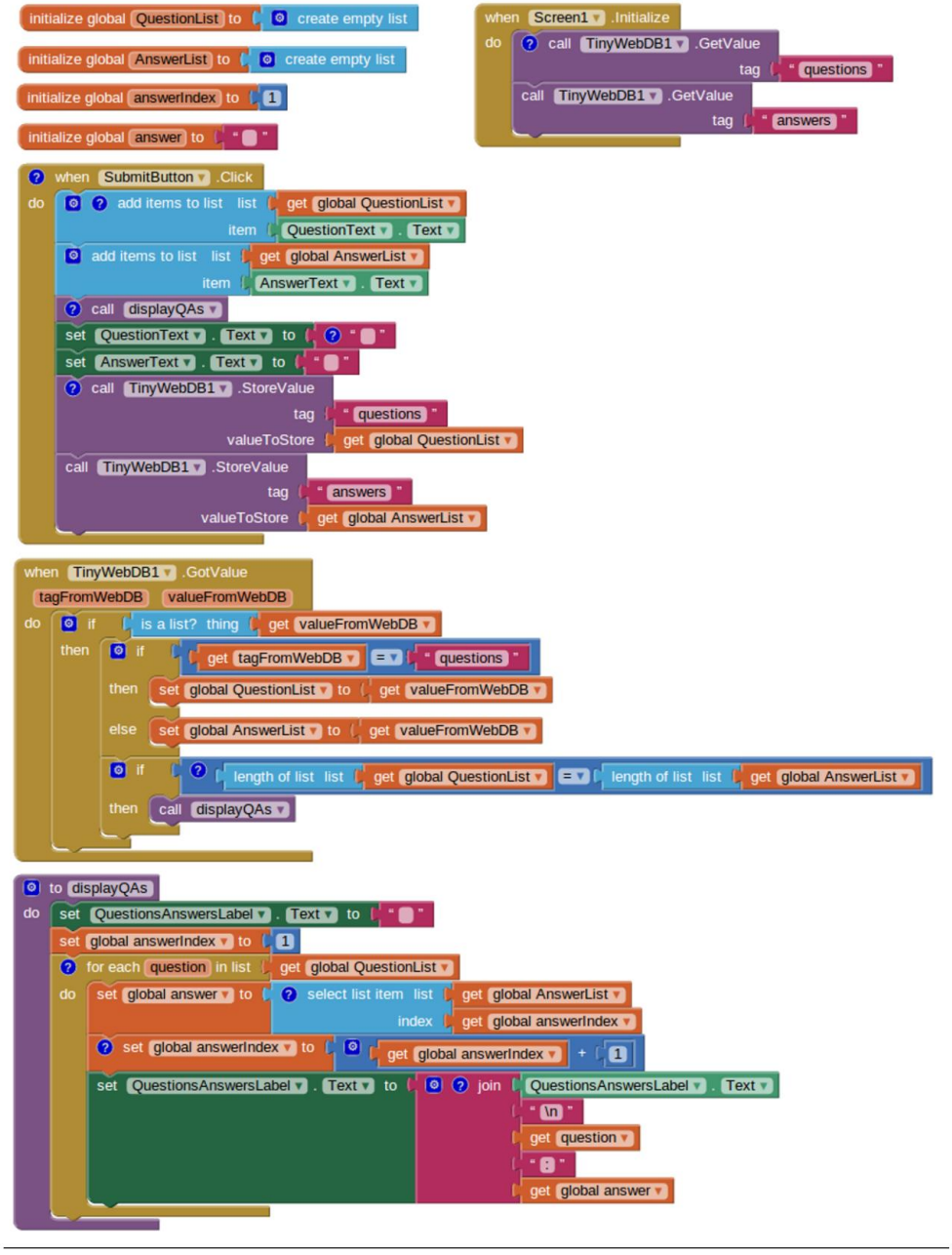
האפליקציה השלמה: MakeQuiz

איור 11-10מציג את הבלוקים עבור כל אפליקציית .MakeQuiz

:TakeQuizאפליקציה לביצוע החידון ב- מאגר מידע

כעת יש לך אפליקציית MakeQuizשתאחסן חידון במסד נתונים אינטרנטי. בניית ,TakeQuiz, האפליקציה שטוענת באופן דינמי את החידון, פשוטה יותר. אתה יכול לבנות אותו עם כמה שינויים בחידון הנשיאים שהשלמת בפרק 8(אם לא השלמת את המדריך הזה, עשה זאת עכשיו לפני שתמשיך).

התחל בפתיחת אפליקציית Presidents Quizשלך ב-ppA ,Inventorבחירה באפשרות ,Save Asו שם הפרויקט החדש ."TakeQuiz"זה ישאיר את אפליקציית Presidents Quizשלך ללא שינוי, אבל עכשיו אתה יכול להשתמש בלוקים שלה כבסיס ל-TziuQekaT.



איור 10-12. הבלוקים עבור MakeQuiz

לאחר מכן, בצע את השינויים הבאים במעצב:

1. גרסה זו לא תציג תמונות עם כל שאלה, אז תחילה הסר את ההפניות לתמונות מאפליקציית Component Designer, ב-TakeQuiz.

בחר כל תמונה מלוח המדיה ומחק אותה. לאחר מכן, מחק את רכיב Image1 , שיסיר את כל ההפניות אליו מערך הבלוקים.

2. מכיוון ש-ziuQekaT יעבוד עם נתוני מסד נתונים שנמצאים ברשת, גרור רכיב TinyWebDB לתוך האפליקציה.

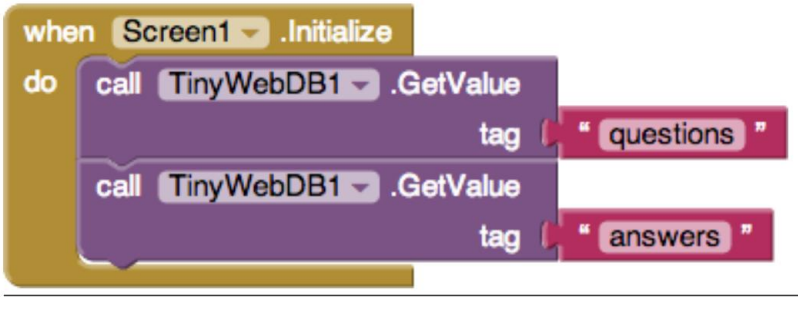
3. מכיוון שאינך רוצה שהמשתמש יענה או ילחץ על NextButton עד שהשאלות ייטענו, בטל את הסימון של המאפיין Enabled של ה- AnswerButton כפתור הבא.

כעת, שנה את הבלוקים כך שהחידון שניתן למשתמש ייטען ממסד הנתונים. ראשית, מכיוון שאין שאלות ותשובות קבועות, הסר את כל בלוקי הטקסט של השאלות והתשובות בפועל מהבלוקים של יצירת רשימה בתוך השאלות והתשובות . הבלוקים המתקבלים צריכים להופיע כפי שמוצג באיור 10-12.

איור 10-13. רשימות השאלות והתשובות מתחילות כעת ריקות

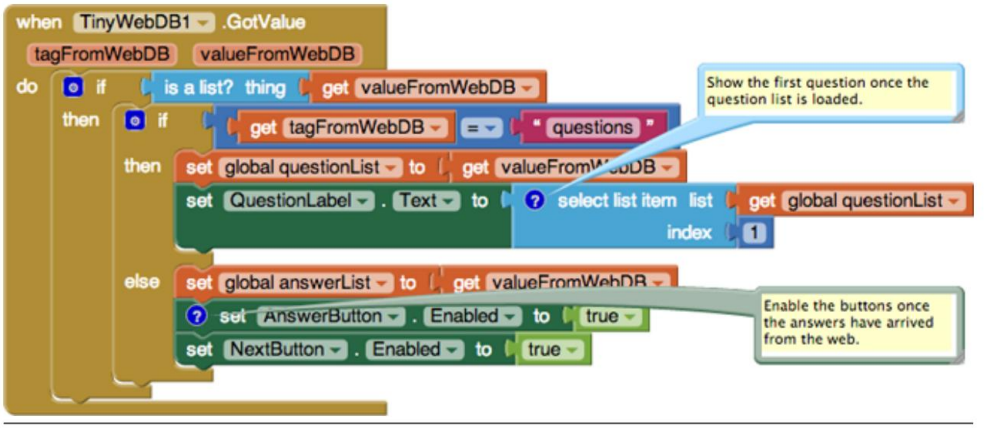
אתה יכול גם למחוק לחלוטין את PictureList; האפליקציה הזו לא תעסוק בתמונות.

כעת, שנה את Screen1.Initialize שלך כך שיקרא ל- TinyWebDB.GetValue פעמיים כדי לטעון את הרשימות, בדיוק כפי שעשית ב-ziuQekaM. הבלוקים צריכים להיראות כפי שהם נראים באיור 10-13.



איור 10-14. בקשת השאלות והתשובות ממאגר האינטרנט

לבסוף, גרור החוצה מטפל באירועים . TinyWebDB.GotValue מטפל באירועים זה אמור להיראות דומה לזה שבו נעשה שימוש ב-ziuQekaM, אבל כאן אתה רוצה להציג רק את השאלה הראשונה ואף אחת מהתשובות. נסה לבצע את השינויים האלה בעצמך תחילה, ולאחר מכן תסתכל על הבלוקים באיור 10-14 כדי לראות אם הם תואמים את הפתרון שלך.



איור 10-15. GotValue מטפל בנתונים המגיעים מהרשת

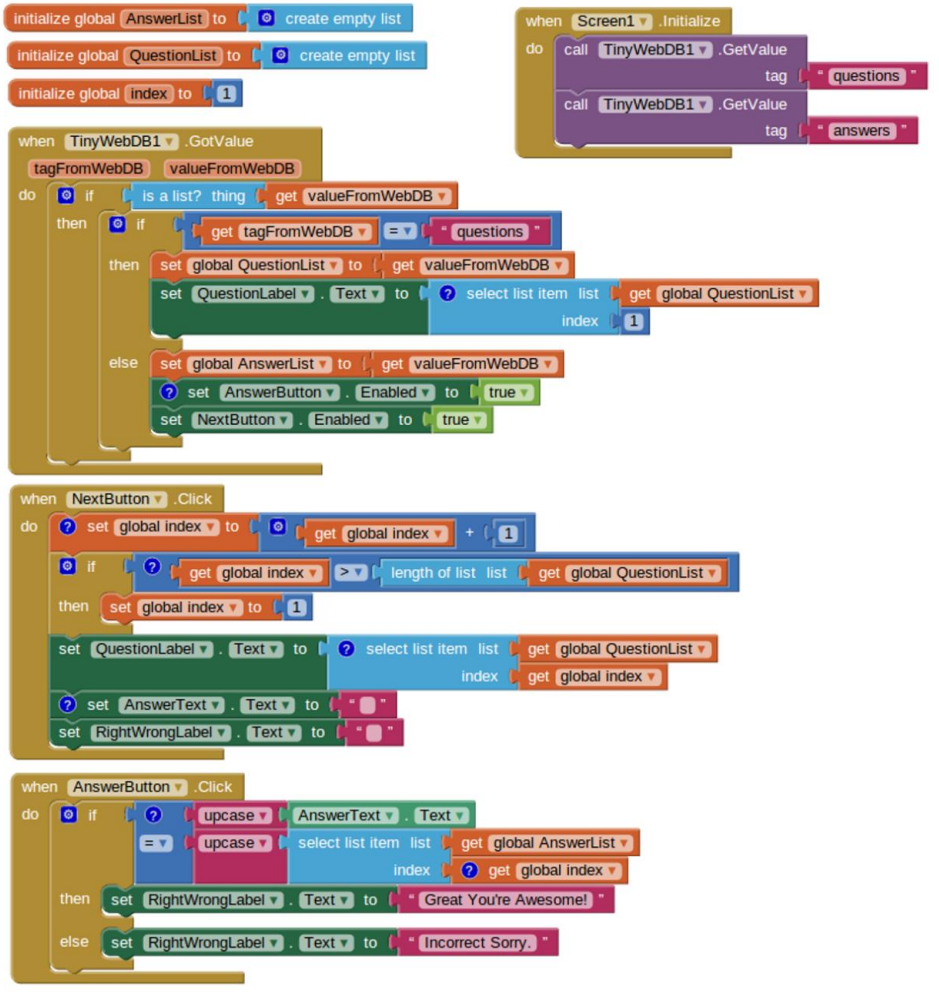
איך הבלוקים עובדים

כאשר האפליקציה מתחילה, Screen1.Initialize מופעל והאפליקציה מבקשת את השאלות והתשובות ממסד הנתונים האינטרנטי. כאשר כל בקשה מגיעה, המטפל באירועים TinyWebDB.GotValue מופעל. האפליקציה תחילה בודקת אם אכן יש נתונים ב-valueFromWebDB באמצעות רשימה? אם היא מוצאת נתונים, האפליקציה שואלת איזו בקשה נכנסה, באמצעות tagFromWebDB, וממקמת את הערך FromWebDB ברשימה המתאימה. אם השאלות נטען, השאלה הראשונה נבחרה מ- QuestionList ומוצגת. אם AnswerList-הנטען, ה- AnswerButton וה- NextButton מופעלים כך שהמשתמש יוכל להתחיל לגשת למבחן.

אלה כל השינויים שאתה צריך עבור TakeQuiz. אם הוספת כמה שאלות ותשובות עם MakeQuiz ואתה מפעיל את TakeQuiz, השאלות שיופיעו צריכות להיות אלו שאתה מזין.

האפליקציה השלמה: TakeQuiz

איור 10-15 מציג את הבלוקים עבור כל אפליקציית TakeQuiz.



איור 10-16. החסימות הסופיות עבור TakeQuiz

וריאציות

לאחר שתגרום ל-ziuQekaM ו-ziuQekaT לעבוד, אולי תרצה לחקור כמה מהווריאציות הבאות:

•אפשר ליוצר החידון לציין תמונה עבור כל שאלה. זה קצת מסובך כי TinyWebDBלא מאפשר לך לאחסן תמונות. לכן, התמונות יצטרכו להיות כתובות URLלתמונות באינטרנט, ויוצר החידון יצטרך להזין כתובות URL אלו כפריט שלישי בטופס .MakeQuizשים לב שאתה יכול להגדיר את המאפיין Pictureשל רכיב תמונה לכתובת .URL

•אפשר ליוצר החידון למחוק פריטים מהשאלות והתשובות. ניתן לאפשר למשתמש לבחור שאלה באמצעות רכיב , ListPickerותוכל להסיר פריט עם בלוק הסר פריט רשימה (זכור להסיר משתי הרשימות ולעדכן את מסד הנתונים). לעזרה עם ListPickerומחיקת רשימה, ראה פרק 19.

•תן ליצרן החידון לתת שם לחידון. יהיה עליך לאחסן את שם החידון תחת תג אחר במסד הנתונים, ותצטרך לטעון את השם יחד עם החידון ב-ziuQekaT. לאחר שטענת את השם, השתמש בו כדי להגדיר את המאפיין Screen.Titleכך שיופיע כאשר המשתמש ניגש לחידון.

•אפשר ליצור חידונים בעלי שם מרובים. תזדקק לרשימה של חידונים, ותוכל להשתמש בכל שם חידון בתור (חלק מה) התג לאחסון השאלות שלו
תשובות.

סיכום

להלן כמה מהמושגים שכיסינו בפרק זה:

•נתונים דינמיים הם מידע שהוזן על ידי משתמש האפליקציה או נטען מתוך א
מאגר מידע. תוכנית שעובדת עם נתונים דינמיים היא מופשטת יותר. למידע נוסף, ראה פרק 19.

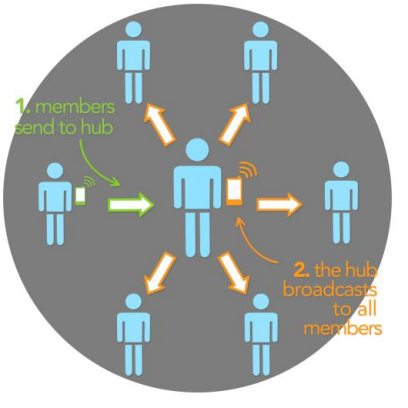
•ניתן לאחסן נתונים באופן קבוע במסד נתונים אינטרנטי באמצעות TinyWebDB
רְכִיב.

•אתה מאחזר נתונים ממסד נתונים של TinyWebDBעל ידי בקשתם באמצעות
.TinyWebDB.GetValueכאשר מסד הנתונים באינטרנט מחזיר את הנתונים, האירוע
TinyWebDB.GotValueמופעל. במטפל האירועים , TinyWebDB.GotValueאתה
יכול לשים את הנתונים ברשימה או לעבד אותם בדרך כלשהי.

•ניתן לשתף נתוני TinyWebDBבין מספר טלפונים ואפליקציות. למידע נוסף על מסדי נתונים
(אינטרנט), ראה פרק 22.

רכזת שידור

איור 11-1.



FrontlineSMSהוא כלי תוכנה המשמש ב
מדינות מתפתחות כדי לפקח על בחירות, לשדר שינויי מזג אוויר ולחבר אנשים שאין להם גישה לאינטרנט אבל יש להם
טלפונים וקישוריות סלולרית. זוהי פרי מוחו של קן בנקס, חלוץ בשימוש בטכנולוגיה ניידת כדי לעזור לאנשים במצוקה.

התוכנה משמשת כמרכז לטקסט SMS
תקשורת בתוך קבוצה. אנשים שולחים קוד מיוחד להצטרפות לקבוצה, ולאחר מכן הם מקבלים הודעות שידור מהרכזת. עבור מקומות ללא גישה לאינטרנט, רכזת השידור יכולה לשמש כחיבור חיוני לעולם החיצון.

בפרק זה, תיצור אפליקציית רכזת שידור הפועלת בדומה ל-FrontlineSMS אבל פועל בטלפון אנדרואיד. עצם הרכזת במכשיר נייד פירושה שהמנהל יכול להיות בתנועה, דבר שחשוב במיוחד במצבים שנויים במחלוקת, כמו מעקב אחר בחירות ומשא ומתן על בריאות.

בפרק זה, תבנה אפליקציית רכזת שידור עבור ריקוד ה-boMhsalF המעשי צוות (TDMF), קבוצה שמשתמשת במרכז כדי לארגן ריקודי אספסוף בכל מקום ובכל זמן. אנשים ירשמו לקבוצה באמצעות הודעת טקסט "joinFMDT"לרכזת, וכל מי שנרשם יכול לשדר הודעות לכל השאר בקבוצה.
האפליקציה שלך תעבד הודעות טקסט שהתקבלו באופן הבא:

1.אם הודעת הטקסט נשלחה ממישהו שעדיין לא נמצא ברשימת השידורים, האפליקציה מגיבה בטקסט שמזמין את האדם להצטרף לרשימת השידורים.

2.אם מתקבלת הודעת הטקסט עם הקוד המיוחד "joinFMDT", האפליקציה מוסיפה השולח לרשימת השידורים.

3.אם הודעת הטקסט נשלחת ממספר שכבר נמצא ברשימת השידורים, ההודעה משודרת לכל המספרים ברשימה.

האפליקציה הזו מסובכת יותר מהאפליקציה ללא טקסט בזמן נהיגה בפרק 4,אבל תבנה אותה פונקציונליות אחת בכל פעם, החל מהתגובה האוטומטית הראשונה

הודעה שמזמינה אנשים להצטרף. עד שתסיים את זה, יהיה לך מושג די טוב כיצד לכתוב אפליקציות תוך שימוש בטקסט SMSכממשק המשתמש. האם אתה רוצה לכתוב אפליקציית הצבעה לפי טקסט כמו אלה המשמשות בתוכניות כישרונות בטלוויזיה, או את אפליקציית ההודעות הקבוצתית הנהדרת הבאה? תלמד איך כאן!

מה תלמד

המדריך מכסה את המושגים הבאים של ממציא אפליקציות, שסביר להניח שחלקם כבר מכירים:

- רכיב הטקסט לשליחת טקסטים ועיבוד טקסטים שהתקבלו.
- רשימת משתנים ונתונים דינמיים -במקרה זה, כדי לעקוב אחר רשימת הטלפון מספרים.
- עבור כל בלוק כדי לאפשר לאפליקציה לחזור על פעולות ברשימת נתונים. במקרה זה, תשתמש עבור כל אחד מהם כדי לשדר הודעות לרשימת מספרי הטלפון.
- רכיב TinyDBלאחסון נתונים באופן מתמשך. המשמעות היא שאם תסגור את האפליקציה ואז תפעיל אותה מחדש, רשימת מספרי הטלפון עדיין תהיה שם.

מתחילים

תזדקק לטלפון עם שירות SMSכדי לבדוק או להפעיל את האפליקציה הזו. תצטרך גם לגייס כמה חברים כדי לשלוח לך הודעות טקסט כדי לבדוק את האפליקציה במלואה.
התחבר לאתר App Inventorוהתחל פרויקט חדש. תן לזה שם ,"BroadcastHub"וגם הגדר את כותרת המסך ל-"tsacdaorB ."Hubלאחר מכן, חבר את המכשיר או האמולטור שלך לבדיקה חיה.

עיצוב הרכיבים

Broadcast Hubמאפשר תקשורת בין טלפונים ניידים. לטלפונים האלה אין צורך להתקין את האפליקציה, או אפילו להיות סמארטפונים; הם יתקשרו באמצעות טקסט עם האפליקציה שלך. אז, במקרה זה, ממשק המשתמש של האפליקציה שלך מיועד רק למנהל הקבוצה.

ממשק המשתמש של המנהל הוא פשוט: הוא מציג את השידור הנוכחי
רשימה; כלומר, רשימת מספרי הטלפון שנרשמו לשירות, וכל הטקסטים שהוא מקבל ומשדר.

כדי לבנות את הממשק, הוסף את הרכיבים המפורטים בטבלה .1-11

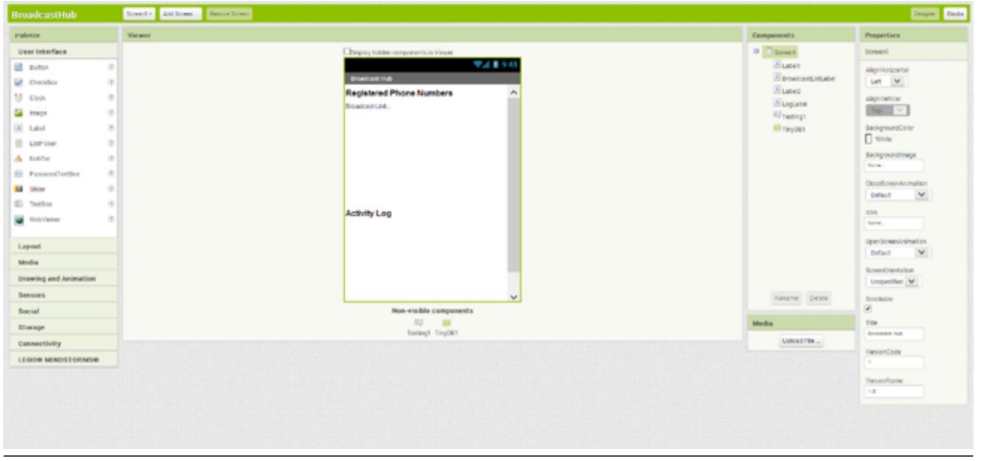
טבלה 11-1. רכיבי ממשק משתמש עבור Broadcast Hub

מַטָרָה	סוג רכיב קבוצת צבעים איך תקרא לזה		
זוהי הכותרת "מספרי טלפון רשומים" למעלה את רשימת מספרי הטלפון.	תווית ממשק משתמש 1		תווית
ממשק משתמש BroadcastListLabelהצג את מספרי הטלפון הרשומים.			תווית
זוהי הכותרת "יומן פעילות" מעל היומן מֵידָע.	תווית ממשק משתמש2		תווית
הצג יומן של הטקסטים שהתקבלו ושודרו.	ממשק משתמש LogLabel		תווית
עבדו את הטקסטים.	חברתי שליחת הודעות טקסט 1		הודעות טקסט
אחסן את רשימת מספרי הטלפון הרשומים.	ממשק משתמש TinyDB1		TinyDB

בעת הוספת הרכיבים, הגדר את המאפיינים הבאים:

1. הגדר את הרוחב של כל תווית ל"מלא הורה" כך שהיא משתרעת על פני הטלפון אופקית.
2. הגדר את גודל הגופן של תוויות הכותרת (Label1 ו- Label2) ל-81 ובדוק את קופסאות FontBold.
3. הגדר את הגובה של BroadcastListLabel ו- LogLabel ל-002 פיקסלים. הם יראו שורות מרובות.
4. הגדר את המאפיין Text של BroadcastListLabel ל-"tsacdaorB List...".
5. הגדר את המאפיין Text של LogLabel לריק .
6. הגדר את המאפיין Text של Label1 ל"מספרי טלפון רשומים".
7. הגדר את המאפיין Text של Label2 ל-"Activity Log".

איור 11-1 מציג את פריסת האפליקציה ב-Component Designer.



איור 11-2. רכזת שידור ב-Components Designer

הוספת התנהגויות לרכיבים

הפעילות עבור Broadcast Hub אינה מופעלת על ידי המשתמש בהקלדת מידע או לחיצה על כפתור; במקום זאת, מדובר בטקסטים המגיעים מטלפונים אחרים. לעבד את הטקסטים הללו ולאחסן את מספרי הטלפון ששלחו אותם ברשימה, תזדקק לדברים הבאים התנהגויות:

- כאשר הודעת הטקסט נשלחת ממישהו שלא נמצא כבר ברשימת השידורים, האפליקציה מגיבה בטקסט שמזמין את השולח להצטרף.
- כאשר הודעת הטקסט "joinFMDT" מתקבלת, רשום את השולח במסגרת רשימת השידורים.
- כאשר הודעת הטקסט נשלחת ממספר שכבר נמצא ברשימת השידורים, ההודעה משודרת לכל המספרים ברשימה.

תגובה לטקסטים נכנסים

תתחיל ביצירת ההתנהגות הראשונה: כאשר אתה מקבל הודעת טקסט, שלח הודעה חזרה לשולחת המזמין אותה להירשם על ידי הודעת טקסט "joinFMDT" בחזרה אליך. אתה תעשה צריך את הבלוקים המפורטים בטבלה 11-2.

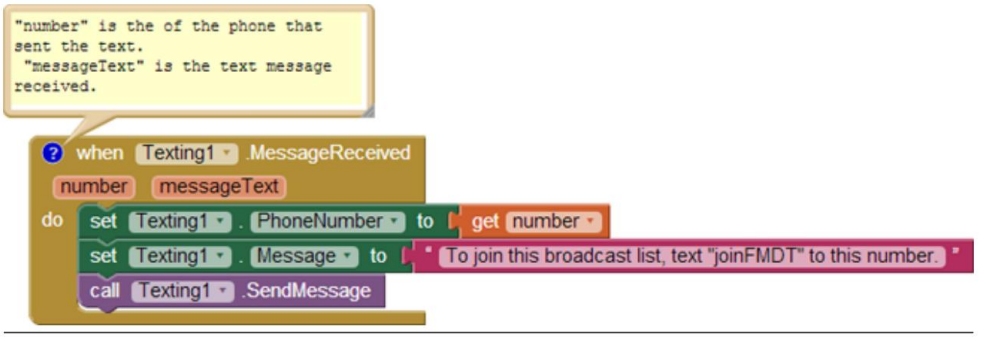
טבלה 11-2. חסימות להוספת הפונקציונליות להזמנת אנשים לקבוצה באמצעות טקסט

מַטָרָה	מְגִרָה	סוג בלוק
מופעל כאשר הטלפון מקבל הודעת טקסט.	שליחת הודעת טקסט	שליחת הודעות .1התקבלה הודעה
הגדר את המספר עבור טקסט ההחזרה.	שליחת הודעת טקסט	הגדר את SMS1.PhoneNumber ל

סוג בלוק	מִגרָה	מַטָרָה
לקבל מספר		גרור מ הטיעון של MessageReceived. אירוע שהתקבל בהודעה זה מספר הטלפון של השולח.
הגדר SMS1.הודעה	שליחת הודעות טקסט	הגדר את הודעת ההזמנה לשליחה.
טקסט ("כדי להצטרף לרשימת השידורים הזו, שלח טקסט 'joinFMDT' למספר הזה")	טֶקסט	הודעת ההזמנה.
שליחת הודעות .1שלח הודעה	שליחת הודעות טקסט	שלח את זה!

איך הבלוקים עובדים

אם השלמת את האפליקציה ללא הודעות טקסט בזמן נהיגה בפרק 4, הבלוקים האלה צריכים נראה מוכר. SMS1.MessageReceived מופעל כאשר הטלפון מקבל כל הודעת טקסט. איור 11-2 מראה כיצד הבלוקים בתוך מטפל האירוע מגדירים את מספר טלפון והודעה של רכיב SMS1 ולאחר מכן שלח את ההודעה.



איור 11-3. שליחת הודעת ההזמנה חזרה לאחר קבלת הודעת טקסט

בדוק את האפליקציה שלך תצטרך טלפון שני כדי לבדוק התנהגות זו; אתה לא רוצה לשלוח הודעה לעצמך, כי זה יכול לנצח! אם אין לך טלפון אחר, אתה יכול להירשם בגוגל קול או שירות דומה ושלח הודעות SMS מאותו שירות לטלפון שלך. מהטלפון השני, שלח את הטקסט "שלום" אל הטלפון שמריץ את האפליקציה. הטלפון השני צריך אז לקבל הודעת טקסט שמזמין אותו להצטרף לקבוצה.

הוספת מספרים לרשימת השידורים

הגיע הזמן ליצור את החסימות להתנהגות השנייה: כאשר הודעת הטקסט "joinFMDT" מתקבל, השולח מתווסף לרשימת השידורים. ראשית, תצטרך הגדר משתנה רשימה, BroadcastList, כדי לאחסן את מספרי הטלפון שנרשמים. מ

את מגירת המשתנים, גרור החוצה בלוק גלובלי אתחול ותן לו שם "רשימת שידורים". אתחול אותו לרשימה ריקה באמצעות יצירת בלוק רשימה מה- מגירת רשימות, כפי שמוצג באיור 11-3(נוסיף את הפונקציונליות לבניית רשימה זו בְּקָרוֹב).

איור 11-4. המשתנה BroadcastListלאחסון רשימת המספרים הרשומים

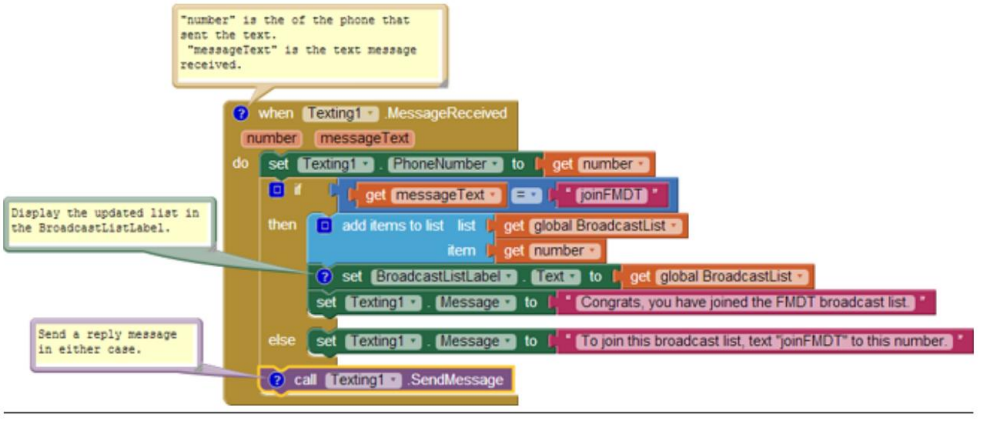
לאחר מכן, שנה את מטפל האירועים Texting1.MessageReceivedכך שיוסיף את מספר הטלפון של השולח ל- BroadcastListאם ההודעה שהתקבלה היא "joinFMDT." תצטרך לחסום אם אחרת כדי לבדוק את ההודעה, ולחסום של הוסף פריט לרשימה הוסף את המספר החדש לרשימה. ערכת הבלוקים המלאה שתזדקק לה רשומה בטבלה 11-3. לאחר הוספת המספר לרשימה, הצג את הרשימה החדשה ב- BroadcastListLabel.

טבלה 11-3. חסימות לבדיקת הודעת טקסט והוספת השולח לרשימת השידורים

מַטָרָה	מְגִרָה	סוג בלוק
בהתאם להודעה שהתקבלה, עשה אחרת דברים.	לִשְׁלוֹט	אחרת
קבע אם messageTextשווה ל "הצטרף "FMDTל	מתמטיקה	=
גרור החוצה אירוע שהתקבל בהודעה חבר את זה לבלוק .= מטפל		קבל הודעהטקסט
חבר את זה לבלוק .=	טֶקסט	טקסט ("הצטרף "FMDTל
הוסף את מספר השולח ל- BroadcastList.	רשימות	הוסף פריטים לרשימה
הרשימה.	גרור החוצה מהמשתנה בלוק אתחול.	קבל את רשימת השידור העולמית
חבר את זה כפריט של הוסף פריטים לרשימה.	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
הצג את הרשימה החדשה.	BroadcastListLabel	הגדר את BroadcastListLabel.Textל
גרור החוצה מהמשתנה BroadcastList. טקסט לחסימה.		רשימת השידור העולמית
הכן הודעות טקסט כדי לשלוח הודעה בחזרה אל שׁוֹלֵחַ.	שליחת הודעות טקסט	הגדר ל-1SMS.הודעה ל
ברך את השולח על הצטרפותו לקבוצה.	טֶקסט	טקסט ("מזל טוב, אתה...")

איך הבלוקים עובדים

שורת הבלוקים הראשונה המוצגת באיור 11-4מגדירה את Texting1.PhoneNumber למספר הטלפון של ההודעה שזה עתה התקבלה; אנחנו יודעים שאנחנו הולכים להגיב לשולח, אז זה קובע את זה. האפליקציה שואלת אם ההודעהtxeT היה הקוד המיוחד, " joinFMDT."אם כן, מספר הטלפון של השולח מתווסף ל- BroadcastList, ונשלחת הודעת מזל טוב. אם ההודעהtxeT היא משהו אחר מאשר " joinFMDT "הודעת התשובה חוזרת על הודעת ההזמנה. לאחר החסימה אם אחרת , נשלחת הודעת התשובה (השורה התחתונה של הבלוקים).



איור .11-5אם ההודעה הנכנסת היא ,"joinFMDT"הוסף את השולח ל- BroadcastList

בדוק את האפליקציה שלך מטלפון שלא מריץ את האפליקציה, שלח את הודעת הטקסט "joinFMDT"לטלפון שמריץ את האפליקציה. אתה אמור לראות את מספר הטלפון הרשום בממשק המשתמש תחת "מספרי טלפון רשומים". הטלפון ה"שולח" אמור לקבל גם את הודעת הברכה כטקסט בתשובה. נסה לשלוח גם הודעה שאינה ,"joinFMDT"כדי לבדוק אם הודעת ההזמנה עדיין נשלחת כהלכה.

שידור ההודעות

לאחר מכן, תוסיף את ההתנהגות כך שהאפליקציה תשדר הודעות שהתקבלו למספרים ב- BroadcastList, אך רק אם ההודעה מגיעה ממספר שכבר מאוחסן ברשימה זו. המורכבות הנוספת הזו תדרוש יותר בלוקי בקרה, כולל עוד בלוקי בקרה ואחד עבור כל אחד. תזדקק לחסימה נוספת אם אחרת כדי לבדוק אם המספר נמצא ברשימה, ולכל חסימה כדי לשדר את ההודעה לכל מספר ברשימה. תצטרך גם להזיז את חוסמי if else המקודם

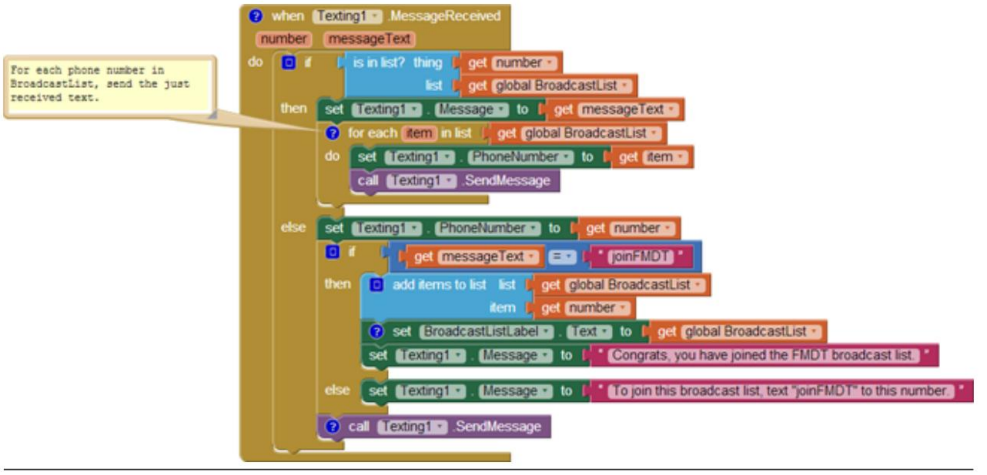
התנהגות ושקע אותם לחלק האחר של החדש אם כן. כל התוספת
בלוקים שתזדקק להם מפורטים בטבלה .11-4

טבלה .11-4חסימות לבדיקה אם השולח כבר בקבוצה

מַטָרָה	מְגִרָה	סוג בלוק
תלוי אם השולח כבר בפנים הרשימה, לעשות דברים שונים.	לִשְׁלוֹט	אחרת
בדוק אם יש משהו ברשימה.	רשימות	נמצא ברשימה?
לחבר את זה לשקע "רשימה" של הוא ברשימה?	גרור החוצה מהמשתנה בלוק אתחול	קבל את רשימת השידור העולמית
לחבר את זה לשקע "דבר" של הוא ברשימה?	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
שלח שוב ושוב הודעה לכל החברים ברשימה.	לִשְׁלוֹט	לכל אחד
חבר את זה לשקע "רשימה" של כל אחד.	גרור החוצה מהמשתנה בלוק אתחול	קבל את רשימת השידור העולמית
הגדר את ההודעה.	שליחת הודעות טקסט 1	הגדר ל-SMS1.הודעה ל
ההודעה שהתקבלה ותהיה מִשׁתָנֶה.	גרור החוצה מה אירוע שהתקבל בהודעה	קבל הודעהטקסט
הגדר את מספר הטלפון.	שליחת הודעות טקסט 1	הגדר את SMS1.PhoneNumber ל
החזק את הפריט הנוכחי ברשימת השידורים; זה מספר (טלפון).	גרור החוצה עבור כל בלוק	לקבל פריט

איך הבלוקים עובדים

האפליקציה הפכה למורכבת מספיק כדי שהיא דורשת בלוק מקונן אם אחרת , אשר
אתה יכול לראות באיור .11-5בלוק מקונן אם אחר הוא אחד שמחובר ל-
שקע של ifחלק של אחר, חיצוני אם אחר. במקרה זה, החיצוני אם אחר
סניף בודק אם מספר הטלפון של ההודעה שהתקבלה כבר נמצא ב-
רשימה. אם כן, ההודעה מועברת לכל מי שברשימה. אם המספר אינו ברשימה,
הבדיקה המקוננת מתבצעת: הבלוקים בודקים אם ההודעהTxeT שווה ל
"הצטרף "FMDTלוהסניף אחת משתי דרכים בהתבסס על התשובה.



איור 11-6. החסימות בודקות אם השולח כבר בקבוצה ומשדרים את ההודעה אם כן

באופן כללי, אם ואם עוד בלוקים ניתן לקנן לרמות שרירותיות, מה שנותן לך את כוח לתכנת התנהגויות מורכבות יותר ויותר (ראה פרק 18 למידע נוסף על בלוקים מותנים).

ההודעה משודרת באמצעות a עבור כל אחד (בתוך הסעיף החיצוני ואז). עבור כל אחד חוזר דרך ברשימת השידור ושולח את ההודעה לכל פריט. כאשר עבור כל חוזר, כל מספר טלפון עוקב מ- BroadcastList מאוחסן בפריט (פריט הוא מציין מיקום משתנה עבור הפריט הנוכחי המעובד בכל אחד). החסימות בתוך עבור כל קבוצה Texting.PhoneNumber לפריט הנוכחי ולאחר מכן לשלוח את ההודעה. למידע נוסף על אופן הפעולה של כל אחת מהן , ראה פרק 20.

בדוק את האפליקציה שלך ראשית, בקש שני טלפונים שונים להירשם על ידי הודעת טקסט "joinFMDT" לטלפון שמריץ את האפליקציה. לאחר מכן, הודעת טקסט נוספת מאחד הטלפונים. שני הטלפונים צריכים לקבל את הטקסט (כולל זה ששלח אותו).

מיפוי של תצוגת הרשימה

האפליקציה יכולה כעת לשדר הודעות, אך ממשק המשתמש של מנהל האפליקציה זקוק לעבודה. ראשית, רשימת מספרי הטלפון מוצגת בצורה לא אלגנטית. באופן ספציפי, כאשר אתה מציב משתנה רשימה בתווית, הוא מציג את הרשימה עם רווחים בין הפריטים, תוך התאמה ככל האפשר בכל שורה. אז, BroadcastListLabel עשוי להציג את BroadcastList כך:

(+1415111-1111 +1415222-2222 +1415333-3333 +1415444-4444)

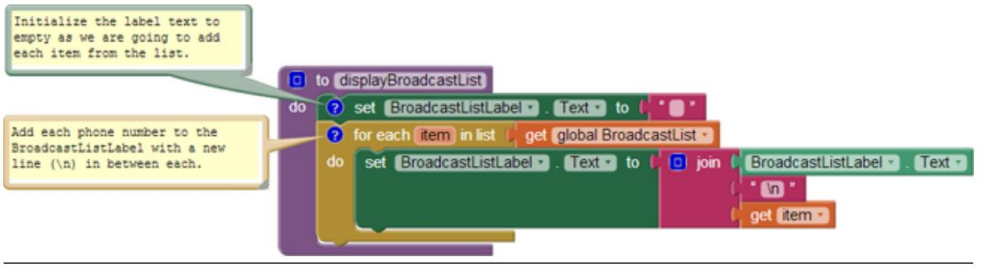
כדי לשפר עיצוב זה, צור הליך בשם displayBroadcastList
באמצעות הבלוקים המפורטים בטבלה .11-5הליך זה מציג את הרשימה עם כל טלפון
מספר בשורה נפרדת. הקפד לקרוא להליך מלמטה כדי להוסיף פריטים
בלוק רשימה כך שהרשימה המעודכנת תוצג.

טבלה .11-5חסימות לניקוי תצוגת מספרי הטלפון ברשימה שלך

מַטָרָה	מְגִרָה	סוג בלוק
צור את ההליך (אל תבחר לבצע את התוצאה).	להליך ("displayBroadcastList")	נהלים
הצג את הרשימה כאן.	הגדר את BroadcastListLabel ל-BroadcastListLabel.Text	
לחץ על טקסט ולאחר מכן לחץ על מחק כדי ליצור אובייקט טקסט ריק.	טקסט ("") טֶקסט	
חזור על המספרים.	לכל אחד לִשְׁלוֹט	
קבל את רשימת השידור העולמית		
שנה זאת עם כל אחד מהמספרים.	הגדר את BroadcastListLabel ל-BroadcastListLabel.Text	
בנה אובייקט טקסט ממספר חלקים.	הצטרף לטקסט פִּקָסטִי	
הוסף את זה לתווית בכל איטרציה של לכל אחד.	BroadcastListLabel	BroadcastListLabel.Text
הוסף תו קו חדש כך הבא המספר נמצא בשורה הבאה.	טֶקסט	טקסט ("\n")
המספר הנוכחי מהרשימה.	גרור החוצה עבור כל בלוק.	לקבל פריט

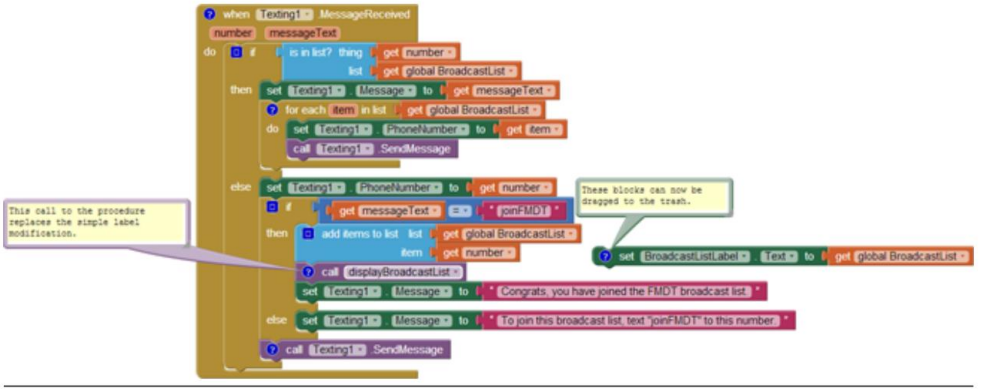
איך הבלוקים עובדים

עבור כל אחד ב- displayBroadcastListמוסיף ברציפות מספר טלפון לסוף
של התווית, כפי שמוצג באיור ,11-6הצבת תו חדש (\n)בין כל
פריט על מנת להציג כל מספר בשורה חדשה.



איור .11-7הצגת מספרי הטלפון עם תו חדש בין כל אחד מהם

כמובן, נוהל displayBroadcastListזה לא יעשה שום דבר אלא אם תתקשר אליו.
התקשר אליו במטפל האירועים , Texting1.MessageReceivedממש מתחת לקריאה כדי
להוסיף פריט לרשימה. הקריאה צריכה להחליף את הבלוקים שפשוט מגדירים את
BroadcastListאתה יכול למצוא את השיחה .BroadcastListLabel.Textל-
displayBroadcastListבלוק במגירה של נהלים.
איור 7-11מראה כיצד החסימות הרלוונטיות בתוך ה- Texting1.MessageReceived
מטפל באירוע צריך להסתכל.



איור .8-11קורא לנוהל displayBroadcastList

למידע נוסף על השימוש בכל אחד מהם כדי להציג רשימה, ראה פרק .20למידע נוסף
מידע על נהלי יצירה והתקשרות, ראה פרק .21

בדוק את האפליקציה שלך הפעל מחדש את האפליקציה כדי לנקות את
הרשימה ולאחר מכן תרשום לפחות שני טלפונים שונים (שוב). האם
מספרי הטלפון מופיעים בקווים נפרדים?

רישום הטקסטים המשודרים

כאשר הודעת טקסט מתקבלת ומשודרת לטלפונים האחרים, האפליקציה צריכה לרשום את
ההתרחשות הזו כדי שמנהל המערכת יוכל לעקוב אחר הפעילות. במעצב הרכיבים הוספת
את התווית LogLabelלממשק המשתמש למטרה זו. כעת, תקודד כמה בלוקים שמשנים את
LogLabelבכל פעם שמגיע טקסט חדש.
אתה צריך לבנות טקסט שאומר משהו כמו "הודעה מ-+1115141-2222 שודרה." המספר
1415111-2222+אינו נתונים קבועים; במקום זאת, זהו הערך של מספר הארגומנט שמגיע
עם האירוע . MessageReceivedאז כדי לבנות את הטקסט, תשרשר את החלק הראשון,
"הודעה מאת", עם בלוק קבל מספר ולבסוף עם החלק האחרון של ההודעה, הטקסט
"שידור".

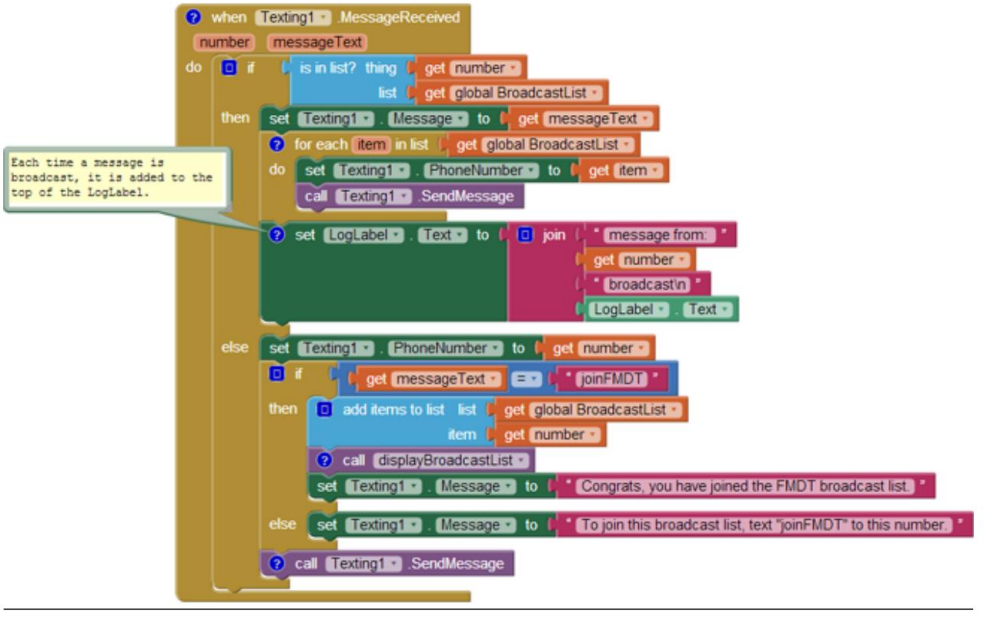
כפי שעשית בפרקים הקודמים, השתמש ב- joinכדי לשרשר את החלקים על ידי שימוש
הבלוקים המפורטים בטבלה 11-6.

טבלה 11-6. חסימות לבניית יומן ההודעות המשודרות שלך

מַטָרָה	מגרה בלוק	
הצג את היומן כאן.		הגדר את lebaLgoL-לLogLabel.Text
בנה אובייקט טקסט ממספר חלקים.	[illegible]	
זו הודעת הדיווח.	[illegible]	טקסט ("הודעה מאת")
מספר הטלפון של השולח.	גרור החוצה אירוע שהתקבל בהודעה מטפל	לקבל מספר
הוסף את החלק האחרון של "הודעה משידור "2222–111 ולכלול קו חדש.	טֶקסט	טקסט ("שידור\n")
הוסף יומן חדש לקודמים.	LogLabel	LogLabel.Text

איך הבלוקים עובדים

לאחר שידור ההודעה שהתקבלה לכל המספרים ב- ,BroadcastListה
האפליקציה משנה כעת את LogLabelכדי להוסיף דוח של הטקסט ששודר זה עתה, כפי שמוצג
באיור 11-8.שימו לב שההודעה מתווספת לתחילת הרשימה במקום
הסוף. בדרך זו, ההודעה האחרונה שנשלחה לקבוצה מופיעה בראש.



איור 11-9. הוספת הודעת שידור חדשה ליומן

בלוק ההצטרפות יוצר ערכים חדשים של הטופס:

הודעה מ: 111-2222שידור

בכל פעם שטקסט משודר, רשומת היומן מתווספת (נוסף מקדימה) ל-
LogLabel.Textכך שהערכים האחרונים יופיעו למעלה. האופן שבו אתה מארגן את בלוק
ההצטרפות קובע את סדר הערכים. במקרה זה, ההודעה החדשה מתווספת עם שלושת
השקעים העליונים של הצטרפות, ו- - LogLabel.Textשמכיל את הערכים הקיימים -מחובר
לשקע האחרון.

ה-"\n" בטקסט "broadcast\n"הוא תו השורה החדשה שגורם לכל יומן
ערך להצגה בשורה נפרדת:

הודעה מ: 1112222שידור
הודעה מ: 555-6666שידור

למידע נוסף על השימוש בכל אחד להצגת רשימה, ראה פרק 20.

אחסון רשימת השידור במסד נתונים

האפליקציה שלך די עובדת, אבל אם סיימתם חלק מהמדריכים המוקדמים יותר, כנראה ניחשתם
שיש בעיה: אם המנהל יסגור את האפליקציה ויפעיל אותה מחדש, רשימת השידורים תאבד וכולם
יצטרכו להירשם שוב.

כדי לתקן זאת, תשתמש ברכיב TinyDBכדי לאחסן ולאחזר את BroadcastListלמסד נתונים
וממנו.

אתה תשתמש בסכימה דומה לזו שבה השתמשת באפליקציית MakeQuiz
(פרק :10)

•אחסן את הרשימה במסד הנתונים בכל פעם שמתווסף פריט חדש.

•כאשר האפליקציה מופעלת, טען את הרשימה ממסד הנתונים למשתנה.

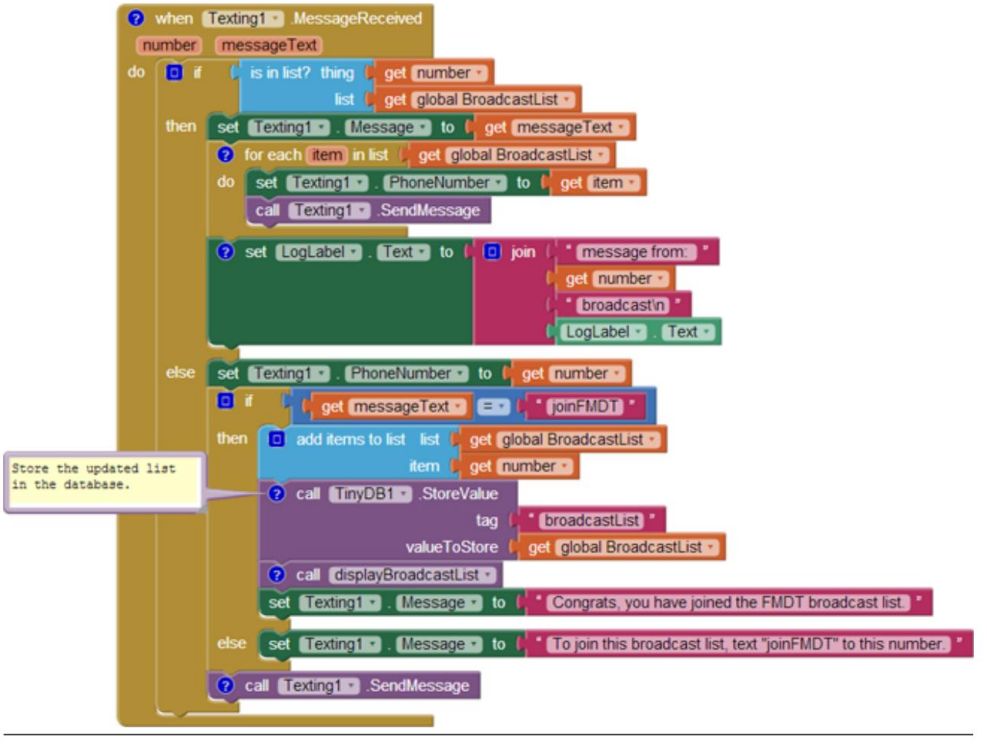
התחל בקידוד הבלוקים המפורטים בטבלה 7-11כדי לאחסן את הרשימה במסד הנתונים.
עם רכיב , TinyDBתג משמש כדי לזהות את הנתונים ולהבדיל אותם מנתונים אחרים
המאוחסנים במסד הנתונים. במקרה זה, אתה יכול לתייג את הנתונים ."broadcastList"-כ
אתה תוסיף את הבלוקים באירוע , Texting1.MessageReceivedמתחת לבלוק הוסף
פריטים לרשימה .

טבלה .7-11חסימות לאחסון הרשימה עם TinyDB

מַטָרָה	מִגרָה	סוג בלוק
אחסן את הנתונים במסד הנתונים.	TinyDB1	TinyDB1.StoreValue
חבר את זה לחריץ "תג" של .StoreValue	טֶקסט	טקסט ("שידור רשימת")
חבר את זה לחריץ "ערך" של .StoreValue	גרור החוצה מבלוק אתחול משתנה	קבל את רשימת השידור העולמית

איך הבלוקים עובדים

כאשר נכנס הודעת טקסט "joinFMDT"ומתווסף למספר הטלפון של החבר החדש הרשימה, TinyDB1.StoreValue נקראת כדי לאחסן את BroadcastListבמסד הנתונים. ה תג (אובייקט טקסט בשם broadcastList) משמש כדי שתוכל מאוחר יותר לאחזר את נתונים. איור 11-9ממחיש שהערך שנקרא על ידי StoreValueהוא המשתנה רשימת שידורים.



איור 11-10. קורא ל-BDyniT כדי לאחסן את רשימת השידורים

טוען את רשימת השידור מבסיס נתונים

הוסף את הבלוקים המפורטים בטבלה 11-8לטעינת הרשימה חזרה בכל פעם שהאפליקציה השקות.

טבלה 11-8. חסימה לטעינת רשימת השידורים חזרה לאפליקציה כאשר היא מופעלת

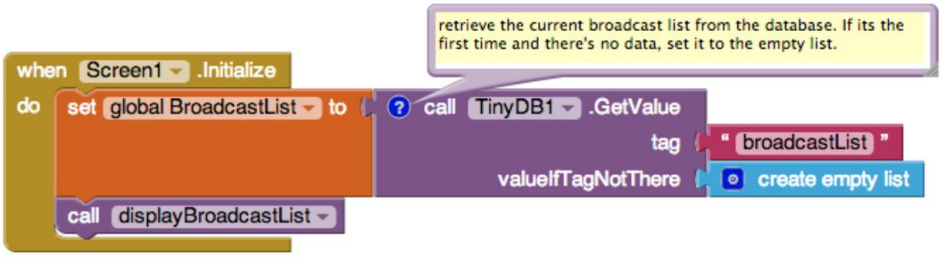
מַטָרָה	מְגֻרָה	סוג בלוק
מופעל כאשר האפליקציה מופעלת.	מסך 1	מסך .1אתחול
בקש את הנתונים ממסד הנתונים.	TinyDB1	TinyDB1.GetValue
חבר את זה לשקע "תג" של GetValue.	טֶקסט	טקסט ("שידור רשימת")

סַעֲרוּלוק	מְגֵרָה	
displayBroadcastList	callנהלים לאחר טעינת הנתונים, הצג אותם.	

כאשר האפליקציה מתחילה, אירוע Screen1.Initializeמופעל, כך שאתה חוסם יכנס למטפל באירוע הזה.

איך הבלוקים עובדים

כאשר האפליקציה מתחילה, האירוע Screen1.Initializeמופעל. הבלוקים המוצגים באיור 11-10מבקשים את הנתונים ממסד הנתונים עם TinyDB1.GetValue.



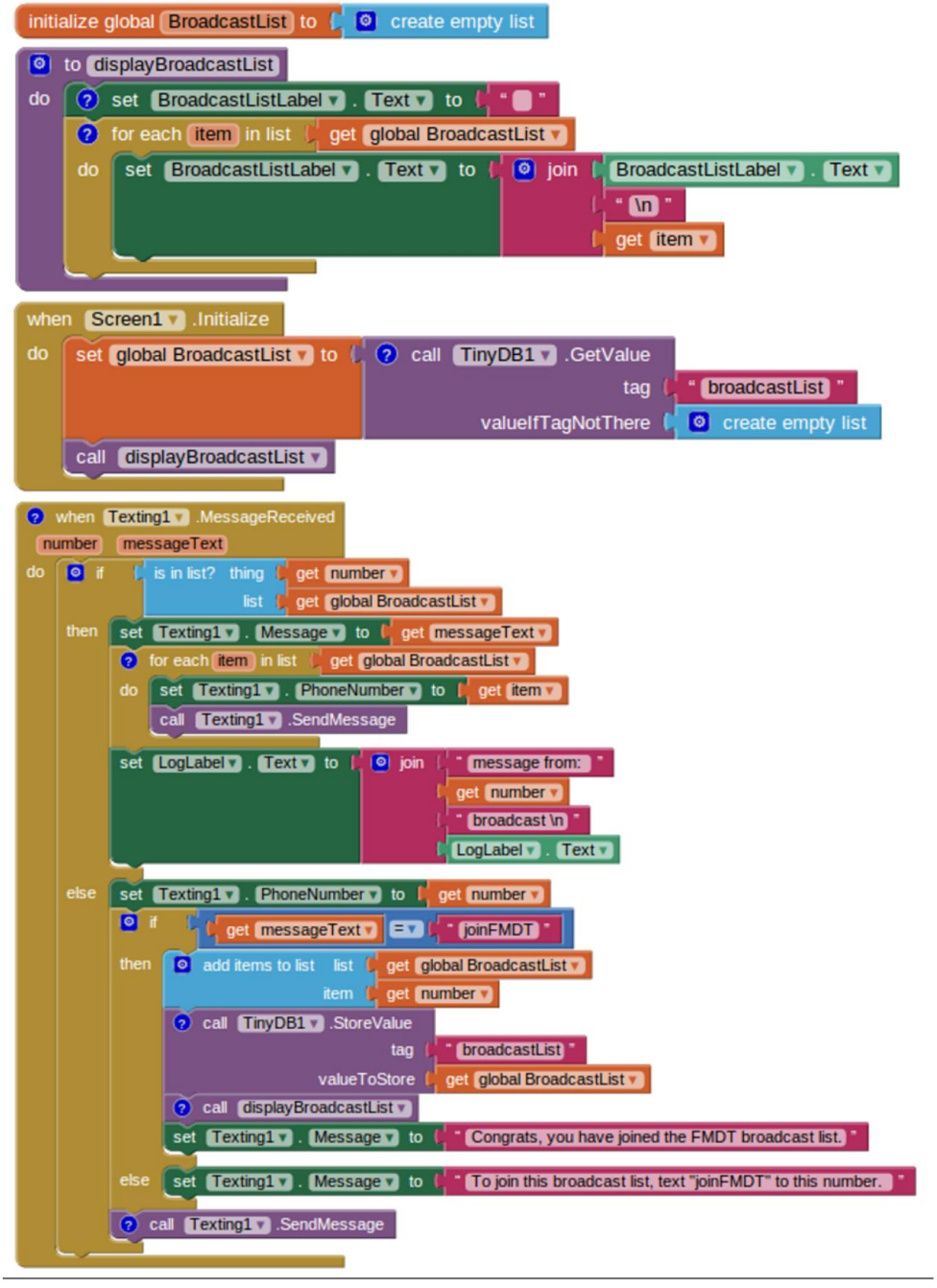
איור .11-11טעינת רשימת השידורים ממסד הנתונים

אתה קורא TinyDB.GetValue-לבאמצעות אותו תג שבו השתמשת לאחסון הרשימה (broadcastList)במקרה הכללי, רשימת מספרי הטלפון המאוחסנת בעבר תוחזר ותמוקם במשתנה BroadcastList. אבל TinyDB.GetValueמספק שקע, valueIfTagNotThere, לציון מה הבלוק צריך להחזיר אם אין עדיין נתונים במסד הנתונים של התג הזה, כפי שיקרה בפעם הראשונה שהאפליקציה הזו תופעל. במקרה זה, מוחזרת רשימה ריקה.

בדוק את האפליקציה שלך אתה יכול להשתמש בבדיקה חיה עבור אפליקציות שמשנות את מסד הנתונים, אך בצע זאת בזהירות. במקרה זה, הוסף טקסט לאפליקציה באמצעות טלפון אחר כדי להוסיף מספרים לרשימת השידורים ולאחר מכן הפעל מחדש את האפליקציה. אתה יכול להפעיל מחדש במצב בדיקה חי על ידי מעבר למעצב ושינוי מאפיין כלשהו, אפילו משהו כמו שינוי הגופן של תווית. שים לב שכדי לבדוק אפליקציות מסד נתונים במלואן, עליך לארוז ולהוריד את האפליקציה לטלפון (בחר "בנה <אפליקציה (שמור apkלמחשב שלי"). לאחר שהורדת את האפליקציה שלך, השתמש בטלפונים האחרים שלך כדי לשלוח הודעת טקסט כדי להצטרף לקבוצה ולאחר מכן לסגור את האפליקציה. אם המספרים עדיין מופיעים בעת הפעלת האפליקציה מחדש, חלק מסד הנתונים עובד.

האפליקציה השלמה: Broadcast Hub

איור 11-11ממחיש את הבלוקים באפליקציית Broadcast Hubשהושלמה.



איור 11-12. אפליקציית Broadcast Hub המלאה

וריאציות

לאחר שחגגת את בניית אפליקציה מורכבת כזו, אולי תרצה לחקור כמה מהווריאציות הבאות:

•האפליקציה משדרת כל הודעה לכולם, כולל הטלפון ששלח את ההודעה. שנה זאת כך שההודעה תשודר לכולם מלבד השולח.

•אפשר לטלפונים של לקוחות להסיר את עצמם מהרשימה על ידי הודעת טקסט "צא" אל אפליקציה. תצטרך להסיר מהרשימה בלוק.

•תן למנהל הרכזת את היכולת להוסיף ולהסיר מספרים מה-רשימת שידורים דרך ממשק המשתמש.

•תן למנהל הרכזת לציין מספרים שאסור להכניס ל-
רשימה.

•התאם אישית את האפליקציה כך שכל אחד יוכל להצטרף כדי לקבל הודעות, אבל רק את מנהל יכול לשדר הודעות.

•התאם אישית את האפליקציה כך שכל אחד יוכל להצטרף לקבלת הודעות, אך רק רשימה קבועה של מספרי טלפון יכולה לשדר הודעות לקבוצה.

סיכום

להלן כמה מהמושגים שכיסינו במדריך זה:

•אפליקציות יכולות להגיב לאירועים שלא יוזמו על ידי משתמש האפליקציה, כגון טקסט מתקבל. זה אומר שאתה יכול לבנות אפליקציות שבהן "המשתמשים" שלך נמצאים בטלפון אחר.

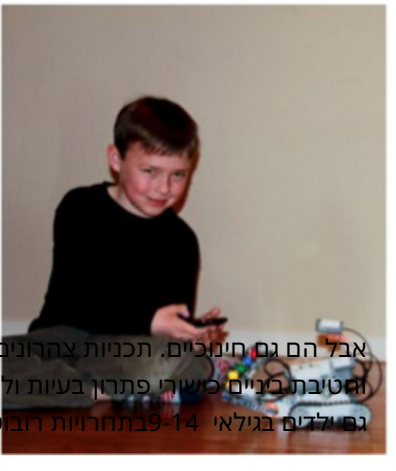
•ניתן להשתמש ב- Nested if elseולכל בלוקים לקוד התנהגויות מורכבות. למידע נוסף על תנאים ולכל איטרציה, ראה פרק 18ופרק ,20בהתאמה.

•ניתן להשתמש בגוש החיבור כדי לבנות אובייקט טקסט ממספר חלקים.

•ניתן להשתמש ב- TinyDBכדי לאחסן ולאחזר נתונים ממסד נתונים. גנרל התוכנית היא לקרוא ל- StoreValueכדי לעדכן את מסד הנתונים בכל פעם שהנתונים משתנים ולהתקשר ל- GetValueכדי לאחזר את נתוני מסד הנתונים כשהאפליקציה מתחילה.

שלט רובוט

איור 12-1.



בפרק זה, תיצור אפליקציה שמסתובבת טלפון אנדרואיד שלך לתוך שלט רחוק עבור רובוט LEGO MINDSTORMS NXT. באפליקציה יהיו כפתורים להנעת הרובוט קדימה ואחורה, פנייה ימינה ושמאלה ועצירה.
אתה תתכנת אותו כך שהרובוט יעצור אוטומטית אם הוא מזהה מכשול. האפליקציה תשתמש ביכולות Bluetooth-השל הטלפון כדי לתקשר עם הרובוט.

רובוטים של LEGO MINDSTORMS כיף לשחק איתם, אבל הם גם חינוכיים. תכניות צהרונים משתמשות ברובוטים כדי ללמד ילדי בית ספר יסודי וחטיבת ביניים כישורי פתרון בעיות ולהכיר להם הנדסה ותכנות מחשבים. רובוטי NXT משמשים גם ילדים בגילאי 14-9 בתחרויות רובוטיקה של ליגת לגו FIRST.

ערכת הרובוטיקה הניתנת לתכנות NXT כוללת יחידה ראשית בשם Intelligent Brick. NXT זה יכול לשלוט על שלושה מנועים וארבעה חיישני קלט. אתה יכול להרכיב רובוט מאלמנטי בנייה של LEGO, גלגלי שיניים, גלגלים, מנועים וחיישנים. הערכה מגיעה עם תוכנה משלה לתכנות הרובוט, אך כעת אתה יכול להשתמש ב-ppA Inventor כדי ליצור אפליקציות אנדרואיד לשליטה ב-TXN באמצעות קישוריות בלוטות'.
האפליקציה בפרק זה מיועדת לעבוד עם רובוט בעל גלגלים וחיישן קולי, כמו רובוט Shooterbot בתמונה כאן. ה-tobretooh S הוא לרוב הרובוט הראשון שאנשים בונים עם סט LEGO MINDSTORMS NXT 2.0. יש לו גלגלים שמאליים המחוברים ליציאת יציאה C, גלגלים ימניים המחוברים ליציאת יציאה B, חיישן צבע המחובר ליציאת קלט 3, וחיישן קולי המחובר ליציאת קלט 4.

מה תלמד

פרק זה משתמש ברכיבים ובמושגים הבאים:

- רכיב BluetoothClient לחיבור ל-TXN

- רכיב ListPickerלספק ממשק משתמש לחיבור ל-NXT
- רכיב NxtDriveלהנעת גלגלי הרובוט
- רכיב NxtUltrasonicSensorלשימוש בחיישן האולטראסוני של הרובוט כדי לזהות מכשולים
- רכיב Notifierלהצגת הודעות שגיאה

מתחילים

תזדקק לאנדרואיד גרסה 2.0ומעלה כדי להשתמש באפליקציה בפרק זה. כמו כן, מטעמי אבטחה, יש להתאים מכשירי Bluetoothלפני שהם יכולים להתחבר זה לזה. לפני שתתחיל לבנות את האפליקציה, תצטרך להתאים את האנדרואיד שלך ל-TXN שלך על ידי ביצוע השלבים הבאים:

1. ב-TXN, לחץ על החץ ימינה עד שיכתוב Bluetoothולאחר מכן הקש על ריבוע כתום.
2. לחץ על החץ ימינה עד להופעת המילה Visibilityולאחר מכן הקש על ריבוע כתום.
3. אם הערך Visibilityכבר נראה, המשך לשלב 4. אם לא, לחץ על השמאל או חץ ימינה כדי להגדיר את הערך לגלוי.
4. באנדרואיד, עבור להגדרות.
שלבים 5-7עשויים להשתנות מעט בהתאם למכשיר האנדרואיד שלך.
5. ודא Bluetooth-שמופעל.
6. לחץ על "Bluetooth"ולאחר מכן על "חפש מכשירים".
7. בקטע "מכשירים זמינים", חפש מכשיר בשם "NXT". אם אי פעם שינית את שם הרובוט שלך, חפש שם מכשיר שתואם את שם הרובוט שלך במקום "NXT".
8. לחץ על "NXT"או על שם הרובוט שלך.
9. ב-TXN, אתה אמור לראות הנחיה עבור מפתח סיסמה. לחץ על הריבוע הכתום כדי לקבל 1234.
10. באנדרואיד, אתה אמור לראות הנחיה עבור ה-NIP. הקלד 1234ולאחר מכן הקש בסדר.
11. הרובוט שלך והאנדרואיד שלך מותאמים כעת.

התחבר לאתר App Inventor בכתובת ai2.appinventor.mit.edu. התחל פרויקט חדש וקרא לו "NXTRemoteControl", והגדר את כותרת המסך ל-"Remote TXN

לִשְׁלוֹט". לחץ על התחבר והגדר את המכשיר (או האמולטור) שלך לבדיקה חיה (ראה http:// /2iaappinventor.mit.edu/explore/setup לקבלת עזרה בהגדרה).

עיצוב הרכיבים

עבור אפליקציה זו, נצטרך ליצור ולהגדיר התנהגויות הן לא גלויות והן גלויות רכיבים.

רכיבים שאינם נראים לעין

לפני יצירת רכיבי ממשק המשתמש, תיצור כמה שאינם גלויים רכיבים, המפורטים בטבלה 12-1ומוצגים באיור ,12-1כדי לשלוט ב-TXN.

טבלה .12-1רכיבים לא גלויים עבור אפליקציית הבקר Robot NXT

		LEGO® MINDSTORMS® NxtDrive	
זיהוי מכשולים.	NxtUltrasonicSensor LEGO® MINDSTORMS® NxtUltrasonicSensor1		
	ממשק משתמש	מודיע1	הצגת הודעות שגיאה.

איור .12-2הרכיבים הלא גלויים המוצגים בתחתית הרכיב מְעַצֵב

הגדר את המאפיינים של הרכיבים בצורה הבאה:

1. הגדר את המאפיין BluetoothClient של NxtDrive1 -ו- NxtUltrasonicSensor1ל- "Bluetooth Client1".

2. סמן את BelowRangeEventEnabled ב- NxtUltrasonicSensor1.

3. הגדר את המאפיין DriveMotors של NxtDrive1:

- אם לרובוט שלך המנוע של הגלגל השמאלי מחובר ליציאת מוצא C ו המנוע של הגלגל הימני מחובר ליציאת פלט B, הגדרת ברירת המחדל של "CB"אין צורך לשנות.
- אם הרובוט שלך מוגדר אחרת, הגדר את המאפיין DriveMotors ל-a ערך טקסט בן שתי אותיות, כאשר האות הראשונה היא יציאת הפלט המחוברת אליה

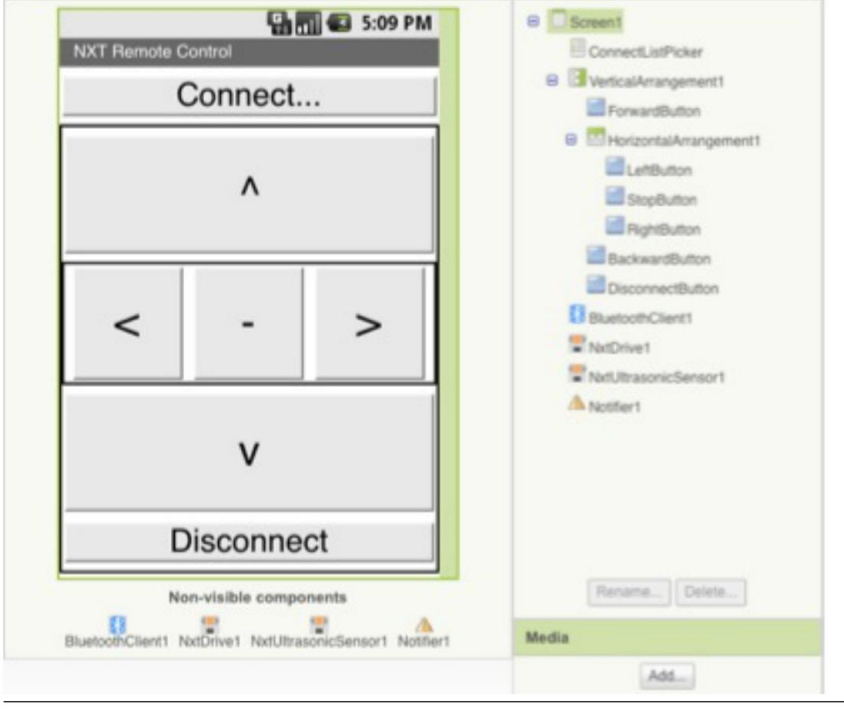
המנוע של הגלגל השמאלי והאות השנייה היא יציאת היציאה המחוברת למנוע של הגלגל הימני.

4. הגדר את המאפיין SensorPort של NxtUltrasonicSensor1.

- אם החיישן האולטראסוני של הרובוט שלך מחובר ליציאת קלט 4, ברירת המחדל אין צורך לשנות את ההגדרה של "4".
- אם הרובוט שלך מוגדר אחרת, הגדר את המאפיין SensorPort ליציאת הקלט המחוברת לחיישן האולטראסוני.

רכיבים גלויים

לאחר מכן, בואו ניצור את רכיבי ממשק המשתמש המוצגים באיור 12-2.



איור 12-3. האפליקציה Component Designer-ב

כדי ליצור את חיבור Bluetooth, התזדקק לכתובת Bluetooth-ההייחודית של ה-TXN. למרבה הצער, כתובות Bluetooth מורכבות משמונה מספרים הקסדצימליים דו ספרתיים (דרך לייצוג ערכים בינאריים) המופרדים על ידי נקודתיים, מה שהופך אותם למסורבלים מאוד להקלדה. לא תרצו להקליד את הכתובת בטלפון בכל פעם שתפעיל את האפליקציה. לכן, כדי להימנע מכך, תשתמש ב- ListPicker המציג רשימה של הרובוטים שהותאמו לטלפון שלך ומאפשר לך לבחור אחד. תשתמש בלחצנים לנסיעה קדימה ואחורה, פנייה ימינה ושמאלה, עצירה ו

מתנתק. אתה יכול להשתמש ב- VerticalArrangement כדי לפרוס הכל למעט
את ,ListPicker וסידור אופקי שיכיל את הכפתורים לפנייה שמאלה,
עוצרים, ופונים ימינה.
אתה יכול לבנות את הממשק המוצג באיור 12-2 על ידי גרירת הרכיבים
המפורטים בטבלה 12-2.

טבלה 12-2. רכיבים גלויים עבור אפליקציית הבקר Robot NXT

סוג רכיב		קְטֵגוֹרִיָה	תֵאוּר
ListPicker		ConnectListPicker	מאפשר למשתמש להתחבר אל
פריסת סידור אנכי		VerticalArrangement1	מיכל ויזואלי.
	לְחִצָן	ForwardButton	ממשק משתמש
		HorizontalArrangement1	מיכל ויזואלי
	ממשק משתמש לחצן שמאלי		
לְחִצָן	ממשק משתמש StopButton		תפסיק.
לְחִצָן	ממשק משתמש לחצן ימין		פנה ימינה.
לְחִצָן	ממשק משתמש BackwardButton		סע אחורה.
לְחִצָן	כפתור ניתוק ממשק משתמש		התנתק מה-TXN.

כדי לסדר את הפריסה החזותית כפי שמוצג באיור 12-2, מקם את לחצן שמאל, כפתור Stop,
ו- RightButton בתוך Horizontal Arrangement1, והצבו , ForwardButton
DisconnectButton -ו HorizontalArrangement1, BackwardButton בפנים
סידור אנכי 1.

הגדר את המאפיינים של הרכיבים באופן הבא:

1. בטל את הסימון שניתן לגלול במסך 1.

2. הגדר את הרוחב של ConnectListPicker ו- DisconnectButton ל-"lliF parent".

3. הגדר את הרוחב והגובה של VerticalArrangement1, ForwardButton ,
HorizontalArrangement1, LeftButton, StopButton, RightButton ו-
כפתור אחורה ל"מלא הורה".

4. הגדר את הטקסט של ConnectListPicker ל-"Connect...".

5. הגדר את הטקסט של ForwardButton ל-"^".

6. הגדר את הטקסט של LeftButton ל-"<".

7. הגדר את הטקסט של כפתור Stop ל-"-".

8. הגדר את הטקסט של לחצן Right ל->.

9. הגדר את הטקסט של כפתור אחורה ל-"v".

10. הגדר את טקסט DisconnectButton ל- "נתק".

11. הגדר את FontSize של ConnectListPicker ו- DisconnectButton ל-03.

12. הגדר את גודל הגופן של RightButton, StopButton, LeftButton , ForwardButton ו-
כפתור אחורה ל-04.

באפליקציה זו, הגיוני להסתיר את רוב ממשק המשתמש עד שה-BluetoothClient יתחבר ל-TXN. כדי להשיג זאת, הגדר את המאפיין Visible של VerticalArrangement1 ל-
hidden. אל דאגה - בעוד רגע נגרום לאפליקציה לחשוף את ממשק המשתמש לאחר שהיא תתחבר ל-TXN.

הוספת התנהגויות לרכיבים

בחלק זה, תתכנת את התנהגות האפליקציה, כולל:

- מתן אפשרות למשתמש לחבר את האפליקציה לרובוט על ידי בחירתה מתוך רשימה.
- מתן אפשרות למשתמש לנתק את האפליקציה מרובוט.
- מתן אפשרות למשתמש להניע את הרובוט באמצעות לחצני הבקרה.
- אילוץ הרובוט לעצור כאשר הוא חש במכשול.

מתחבר ל-TXN

ההתנהגות הראשונה שתוסיף היא התחברות ל-TXN. כאשר תלחץ על ConnectListPicker, הוא יציג רשימה של הרובוטים המזווגים. כאשר אתה בוחר רובוט, האפליקציה תיצור חיבור בלוטות' לרובוט זה.

הצגת רשימת הרובוטים

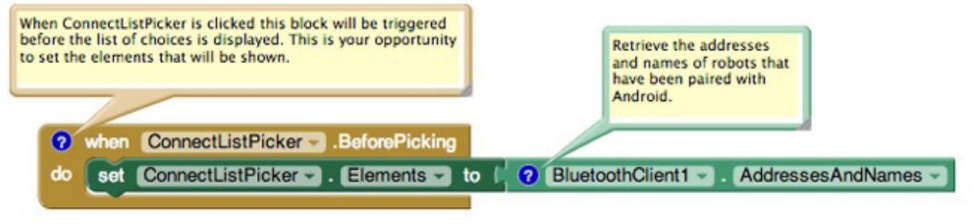
כדי להציג את רשימת הרובוטים, תשתמש ב- ConnectListPicker. ListPicker נראה כמו כפתור, אך כאשר לוחצים עליו, הוא מציג רשימה של פריטים שמהם אתה יכול לבחור אחד. תשתמש בגוש BluetoothClient1.AddressesAndNames כדי לספק רשימה של הכתובות והשמות של מכשירי Bluetooth שהותאמו לאנדרואיד.
מכיוון ש- BluetoothClient1 משמש עם רכיבי NXT, הוא מגביל אוטומטית את ההתקנים הכלולים במאפיין AddressesAndNames לאלה שהם רובוטים, כך שלא תראה סוגים אחרים של התקני Bluetooth (כגון אוזניות) ברשימה. טבלה 12-3 מפרטת את הבלוקים שתזדקק לשלב זה.

טבלה .12-3חסימות להוספת ListPickerלאפליקציה

סוג בלוק	מִגזָרָה	מַטָרָה
	ConnectListPicker	מופעל כשלוחצים על ConnectListPicker. ConnectListPicker.BeforePicking
		הגדר את ConnectListPicker ל-ConnectListPicker.Elements הגדר את האפשרויות שיופיעו.
BluetoothClient1.AddressesAndNames	BluetoothClient1	הכתובות והשמות של רובוטים שהותאמו לאנדרואיד.

איך הבלוקים עובדים

כשלוחצים על ConnectListPicker, אירוע ConnectListPicker.BeforePickingמופעל לפני הצגת רשימת האפשרויות, כפי שמוצג באיור .12-3כדי לציין את הפריטים שיוצגו, הגדר את המאפיין ConnectListPicker.Elementsלבלוק ConnectListPicker יפרט את הרובוטים BluetoothClient1.AddressesAndNames . שהותאמו לאנדרואיד.



איור .12-4מציג את רשימת הרובוטים

בדוק את האפליקציה שלך בטלפון שלך, לחץ על "התחבר..." וראה מה קורה. אתה אמור לראות רשימה של כל הרובוטים שהטלפון שלך הוצמד אליהם. אם אתה רק רואה מסך שחור, הטלפון שלך לא הותאם לאף רובוט. אם אתה רואה כתובות ושמות של התקני Bluetoothאחרים, כגון אוזניות ,Bluetoothהמאפיין BluetoothClientשל NxtDrive1 ו-1rosneScinosartlUtxN לא הוגדר כהלכה.

ביצוע חיבור BLUETOOTH-ה

לאחר שתבחר רובוט מהרשימה, האפליקציה תתחבר לרובוט זה באמצעות .Bluetoothאם החיבור יצליח, ממשק המשתמש ישתנה.
ConnectListPickerיוסתר, ושאר רכיבי ממשק המשתמש יופיעו. אם הרובוט לא מופעל, החיבור ייכשל ותופיע הודעת שגיאה. תשתמש בגוש BluetoothClient1.Connectכדי ליצור את החיבור. ה

מאפיין ConnectListPicker.Selection מספק את הכתובת והשם של הנבחר
רובוט. תשתמש ב- if then block כדי לבדוק אם החיבור הצליח. נו
הוסף עוד לבלוק , if then כך שיהיו לו שלושה אזורים שונים שבהם נמצאים בלוקים
מחובר: אם, אז, ועוד . אזור if יכיל את ה- BluetoothClient1.Connect
לַחסוֹם. האזור אז יכיל את הבלוקים שיבוצעו אם החיבור הוא
מוצלָח. אזור אחר יכיל את הבלוקים שיבוצעו אם החיבור נכשל.

אם החיבור הצליח, תשתמש במאפיין Visible כדי להסתיר
ConnectListPicker והצג את ,VerticalArrangement1 המכיל את שאר ה-
רכיבי ממשק משתמש. אם החיבור נכשל, תשתמש ב-
בלוק Notifier1.ShowAlert להצגת הודעת שגיאה. טבלה 12-4 מפרטת את הבלוקים
תצטרך עבור ההתנהגות הזו.

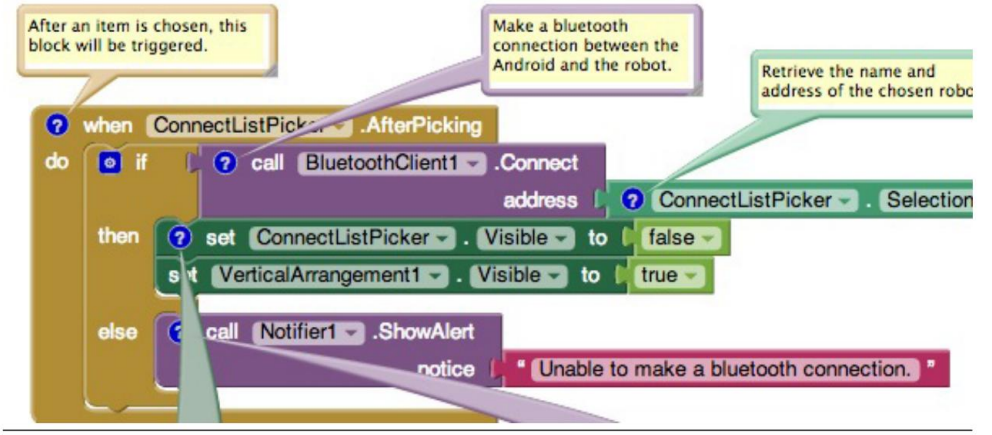
טבלה 12-4. חסימות לשימוש Bluetooth- בכדי להתחבר לרובוט

מַטָרָה	מְגִרָה	סוג בלוק
מופעל כאשר בוחרים רובוט ConnectListPicker.	ConnectListPicker	ConnectListPicker.AfterPicking
בדוק אם חיבור Bluetooth-ה מוצלָח.	לְשָׁלוֹט	אם אז
התחבר לרובוט.	BluetoothClient1	BluetoothClient1.Connect
הכתובת והשם של הרובוט הנבחר.	ConnectListPicker	ConnectListPicker.Selection
הסתר ConnectListPicker.	ConnectListPicker	הגדר יולG.ConnectListPicker ל
חבר לסט ConnectListPicker. גלוי ל. הגיון		שֶׁקֶר
VerticalArrangement1 הצג את שאר ממשק המשתמש.		מַעֲרֶכֶת סידור אנכי .1 גלוי ל
חבר לסט VerticalArrangement1. גלוי ל. הגיון		נָכוֹן
נוטיפר1 הצג הודעת שגיאה.		Notifier1.ShowAlert
הודעת השגיאה.	טֶקסט	טקסט "לא ניתן ליצור בלוטות' חיבור."

איך הבלוקים עובדים

לאחר בחירת רובוט, מופעל אירוע , ConnectListPicker.AfterPicking כמו
מוצג באיור 12-4. בלוק BluetoothClient1.Connect יוצר את Bluetooth-ה
חיבור לרובוט הנבחר. אם החיבור מצליח, החסימות אז כן
מבוצע: המאפיין ConnectListPicker.Visible מוגדר ל-eslaf כדי להסתיר
ConnectListPicker, והמאפיין VerticalArrangement1.Visible מוגדר כ-eurt to

הצג את ,VerticalArrangement1המכיל את לחצני השלט הרחוק. אם ה
החיבור נכשל, החסימות האחרות מבוצעות: הבלוק Notifier1.ShowAlert
מציג הודעת שגיאה.



איור .5-12ביצוע חיבור בלוטות'

ניתוק מה-TXN

אתה כנראה מתלהב מחיבור האנדרואיד שלך ל-TXN שלך, אבל לפניך
עשה זאת, בוא נעשה דבר נוסף: להוסיף את ההתנהגות לניתוק. ככה, אתה תצליח
להיות מסוגל לבדוק גם חיבור וגם ניתוק. כאשר DisconnectButtonהוא
לחצן, האפליקציה תסגור את חיבור Bluetooth-הוממשק המשתמש יסגור
שינוי. ConnectListPickerיופיע שוב ושאר ממשק המשתמש
רכיבים יוסתרו. השתמש בלוקים המפורטים בטבלה 5-12כדי לבנות את
. 1BluetoothClientחסימת ניתוק שסוגרת את חיבור .Bluetooth-האתה

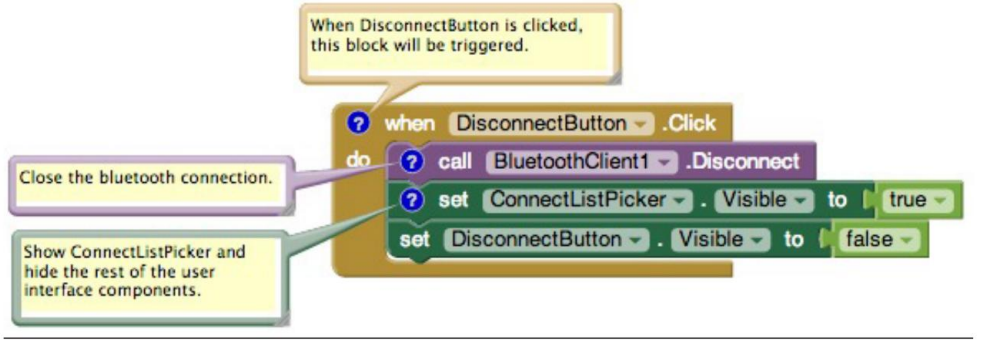
השתמש במאפיין Visibleכדי להציג את ConnectListPickerולהסתיר
,VerticalArrangement1המכיל את שאר רכיבי ממשק המשתמש.

טבלה .5-12בלוקים לניתוק מהרובוט

מַטָרָה	מְגֶרָה	סוג בלוק
מופעל בעת לחיצה על הכפתור .Disconnect	כפתור ניתוק	DisconnectButton.Click
התנתק מהרובוט.	BluetoothClient1	BluetoothClient1.נתק
הצג את .ConnectListPicker	ConnectListPicker	הגדר את ConnectListPicker.Visible ל
חבר לסט .ConnectListPickerגלוי ל.	הִגָיוֹן	נָכוֹן

סוג בלוק	מְגֻרָה	מַטָרָה
מַעֲרֶכֶת ל	הסתר את שאר ממשק המשתמש. סידור אנכי 1. גלוי	VerticalArrangement1
שֶׁקֶר	הִגָיוֹן	חבר לסט 1. VerticalArrangement גלוי ל.

כשלוחצים על DisconnectButton, האירוע DisconnectButton.Clicked הוא מופעל, כפי שמוצג באיור 12-5. בלוק BluetoothClient1.Disconnect סוגר את חיבור בלוטות'. המאפיין ConnectListPicker.Visible מוגדר כ-eurt כדי להראות ConnectListPicker, והמאפיין VerticalArrangement1.Visible מוגדר כ-eslaf ל הסתר את VerticalArrangement1, המכיל את לחצני השלט הרחוק.



איור 12-6. ניתוק מהרובוט

בדוק את האפליקציה שלך ודא שהרובוט שלך מופעל ואז, בטלפון שלך, לחץ על "התחבר..." ובחר את הרובוט שאתה רוצה להתחבר ל. ייקח רגע לעשות את חיבור בלוטות'. לאחר שהרובוט מתחבר, אתה אמור לראות הכפתורים לשליטה ברובוט, כמו גם Disconnect-ה לַחְצָן. לחץ על הלחצן נתק. הכפתורים עבור השליטה ברובוט צריכה להיעלם, וה-tcennoC הכפתור אמור להופיע שוב.

נוהג ב-TXN

בואו נגיע לחלק המהנה באמת: הוספת התנהגות לנהיגה קדימה ואחורה, פונים ימינה ושמאלה, ועוצרים. אל תשכח לעצור -אם תעשה זאת, תעשה זאת

יש רובוט שיצא משליטה על הידיים! רכיב NxtDriveמספק fve
בלוקים להנעת המנועים של הרובוט:

MoveForward •ללא הגבלת זמן מניע את שני המנועים קדימה.

MoveBackward •מניע את שני המנועים לאחור ללא הגבלת זמן.

•סיבוב נגד השעון הופך את הרובוט שמאלה ללא הגבלת זמן על ידי הנעת ה-
מנוע ימני קדימה והמנוע השמאלי אחורה.

•סיבוב בכיוון השעון הופך את הרובוט ימינה ללא הגבלת זמן על ידי נסיעה שמאלה
מנוע קדימה והמנוע הימני אחורה.

•עצירה מפסיקה את שני המנועים.

לבלוקים ...Moveו- ... Turnלכל אחד מהם יש פרמטר שנקרא .Powerאתה תשתמש ב-a
בלוק מספרים, יחד עם כל שאר הפריטים המפורטים בטבלה ,12-6כדי לציין את
כמות הכוח שהרובוט צריך להשתמש כדי לסובב את המנועים. הערך יכול לנוע בין 0
ל-001. עם זאת, אם תציין כוח קטן מדי, המנועים ישמיעו קול יבבה
אבל לא להסתובב. ביישום זה, תשתמש ב-09 (אחוזים).

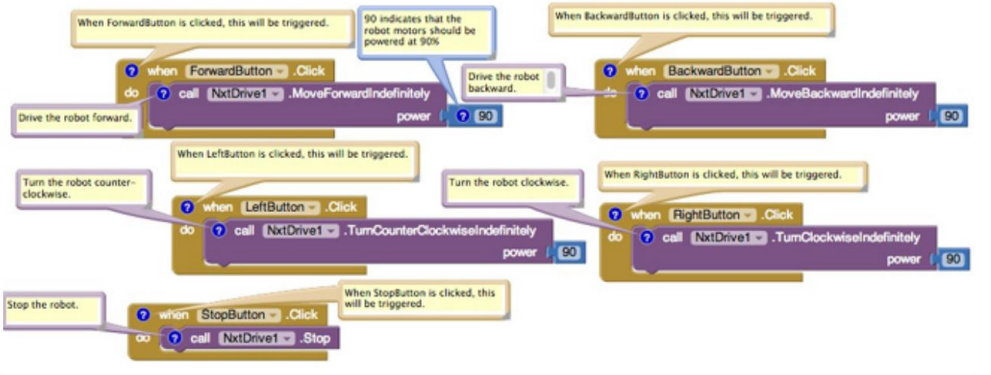
טבלה .12-6בלוקים לשליטה ברובוט

מַטָרָה	מְגִרָה	סוג בלוק
מופעל כאשר ForwardButtonהוא לחצו.	כפתור קדימה	ForwardButton.Click
הסע את הרובוט קדימה.	NxtDrive1	NxtDrive1.MoveForwardInfinitely
כמות הכוח.	מתמטיקה	מספר (90)
מופעל כאשר BackwardButtonהוא לחצו.	כפתור אחורה	לחצן אחורה. לחץ
הסע את הרובוט לאחור.	NxtDrive1	NxtDrive1.MoveBackwardInfinitely
כמות הכוח.	מתמטיקה	מספר (90)
מופעל כאשר LeftButton לחצו.	לחצן שמאל	לחצן שמאלי. לחץ
סובב את הרובוט נגד כיוון השעון.	NxtDrive1	NxtDrive1. TurnNxtDrive1נגד כיוון השעון ללא הגבלת זמן
כמות הכוח.	מתמטיקה	מספר (90)
מופעל כאשר RightButton לחצו.	כפתור ימני	לחצן ימני. לחץ
סובב את הרובוט בכיוון השעון.	NxtDrive1	NxtDrive1.TurnClockwiseInfinitely
כמות הכוח.	מתמטיקה	מספר (90)

סוג בלוק	מְגֻרָה	מַטָרָה
StopButton.Click	כפתור עצור	מופעל כאשר StopButton לחצו.
NxtDrive1.Stop	NxtDrive1	עצור את הרובוט.

איך הבלוקים עובדים

כאשר לוחצים על ,ForwardButton האירוע ForwardButton.Clicked מופעל. ה
בלוק NxtDrive1.MoveForwardIndefinitely המוצג באיור 12-6 משמש להזזת
רובוט קדימה בהספק של .90%שאר האירועים פועלים באופן דומה עבור האחר
כפתורים, כל אחד מהם מפעיל את הרובוט אחורה, שמאלה וימינה.



איור .12-7נוהגים ברובוט

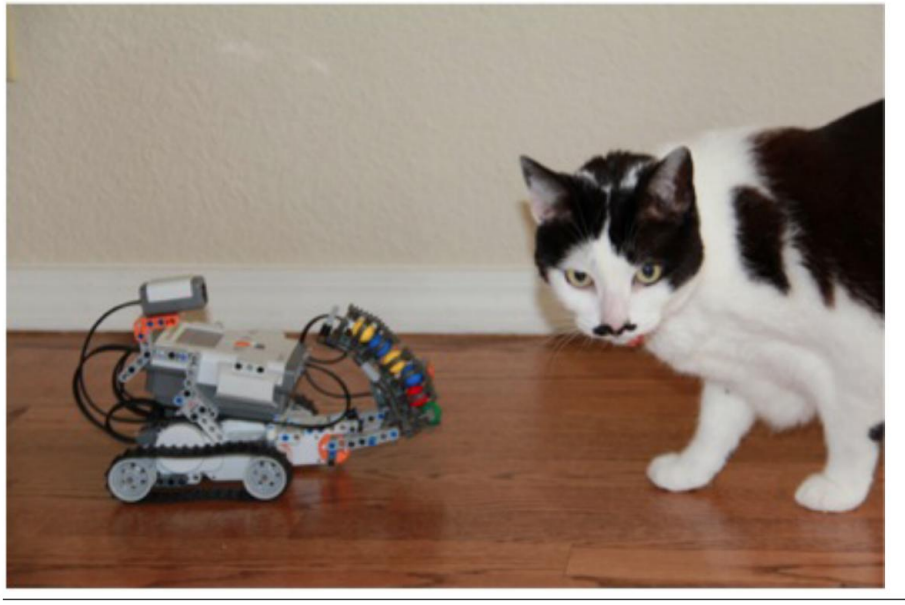
כאשר לוחצים על ,StopButton האירוע StopButton.Clicked מופעל. ה
בלוק NxtDrive1 . Stop משמש לעצירת הרובוט. בדוק את האפליקציה שלך. פעל לפי ההוראות
בקטע הקודם "בדוק את האפליקציה שלך" כדי להתחבר ל-TXN. ודא שהרובוט כן
לא על שולחן שבו הוא עלול ליפול, ולאחר מכן בדוק את התנהגותו באופן הבא:

1. לחץ על כפתור קדימה. הרובוט צריך להתקדם.
2. לחץ על כפתור אחורה. הרובוט צריך לנוע אחורה.
3. לחץ על הכפתור השמאלי. הרובוט צריך להסתובב נגד כיוון השעון.
4. לחץ על הכפתור הימני. הרובוט צריך להסתובב עם כיוון השעון.
5. לחץ על כפתור העצירה. הרובוט צריך לעצור.

אם הרובוט שלך לא זז, אבל אתה יכול לשמוע קול יבבה, ייתכן שיהיה עליך לעשות זאת
להגדיל את הכוח. אתה יכול להשתמש ב-001 להספק מרבי.

שימוש בחיישן האולטראסוני לזיהוי מכשולים

באמצעות החיישן האולטראסוני, הרובוט יעצור אם הוא נתקל במכשול בטווח של 30 סנטימטרים, כגון החסימה המוצגת באיור 12-7.
אתה יכול להשתמש ברכיב NxtUltrasonicSensor כדי לזהות מכשולים. יש לו שני מאפיינים בשם BottomOfRange ו- TopOfRange המגדירים את טווח הזיהוי בסנטימטרים. כברירת מחדל, המאפיין BottomOfRange מוגדר ל-03 סנטימטרים ו- TopOfRange מוגדר ל-09 סנטימטרים.



איור 12-8. מכשול ביתי נפוץ לרובוט ה-TXN שלך

לרכיב NxtUltrasonicSensor יש גם שלושה אירועים הנקראים BelowRange, WithinRange, ו- AboveRange. אירוע BelowRange יופעל כאשר יתגלה מכשול במרחק מתחת ל-BottomOfRange. האירוע WithinRange יופעל כאשר יתגלה מכשול במרחק בין BottomOfRange ל- TopOfRange. אירוע AboveRange יופעל כאשר מכשול מזוהה במרחק מעל TopOfRange.

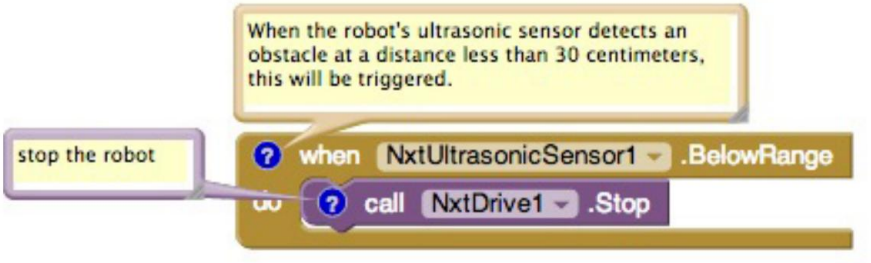
תשתמש בבלוק האירועים NxtUltrasonicSensor1.BelowRange, המוצג בטבלה 12-7, כדי לזהות מכשול בטווח של 30 סנטימטרים. אם אתה רוצה לזהות מכשול במרחק אחר, אתה יכול להתאים את המאפיין BottomOfRange. תשתמש בגוש NxtDrive1.Stop כדי לעצור את הרובוט.

טבלה 12-7. בלוקים לשימוש ב-NxtUltrasonicSensor

סוג בלוק	מְגִרָה	מַטָרָה
NxtUltrasonicSensor1.BelowRange במרחק מתחת ל-03 סנטימטרים.	NxtUltrasonicSensor1	מופעל כאשר החיישן האולטראסוני מזהה מכשול
NxtDrive1.Stop	NxtDrive1	עצור את הרובוט.

איך הבלוקים עובדים

כאשר החיישן הקולי של הרובוט מזהה מכשול במרחק מתחת ל-03 סנטימטרים, האירוע NxtUltrasonicSensor1.BelowRange מופעל, כפי שמוצג באיור 12-8. בלוק NxtDrive1.Stop עוצר את הרובוט.



איור 12-9. זיהוי מכשול ועצירת הרובוט

בדוק את האפליקציה שלך עקוב אחר ההוראות בסעיף הקודם "בדוק את האפליקציה שלך" כדי להתחבר ל-TXN. באמצעות לחצני הניווט, הסע את הרובוט שלך לעבר מכשול, כגון חתול.
הרובוט צריך לעצור כאשר הוא מתקרב למרחק של 30 סנטימטרים מהחתול.

אם הרובוט לא עוצר, ייתכן שהחתול התרחק מהרובוט לפניו
צייר בטווח של 30 סנטימטרים. ייתכן שתצטרך לבדוק את האפליקציה שלך עם מכשול דומם.

וריאציות

לאחר שתגרום ליישום הזה לעבוד - ובילה מספיק זמן במשחק עם רובוט ה-TXN שלך - אולי תרצה לנסות את הדברים הבאים:

- שנה את כמות הכוח בעת הפעלת הרובוט.

- תוכל לעשות זאת על ידי שינוי הערך המספרי שאתה מחבר ל-
  MoveBackwardInfinitely, TurnCounterwiseInfinitely,
  MoveForwardIndefinitely, ו-TurnClockwiseIndefinitely חסימות.

- השתמש ב- NxtColorSensor כדי להאיר באור אדום כאשר מזוהה מכשול.
  - ניתן להשתמש ברכיב NxtColorSensor וב- GenerateColor שלו
    תכונה.
  - יהיה עליך להגדיר את המאפיין DetectColor ל-eslaf (או לבטל את הסימון
    שלו (Component Designer-במכיוון שחיישן הצבע אינו יכול לזהות וליצור
    צבע בו-זמנית.
  - השתמש ב- OrientationSensor כדי לשלוט ברובוט.
  - השתמשו ברכיבי בנייה של LEGO כדי לחבר פיזית את הטלפון לרובוט.
    צור יישומים שהופכים את הרובוט לאוטונומי.

סיכום

להלן כמה מהמושגים שכיסינו במדריך זה:

- אתה יכול להשתמש ברכיב ListPickerכדי לבחור מתוך רשימה של רובוטים מזווגים.
- רכיב BluetoothClientיוצר את החיבור לרובוט.
- הרכיב Notifierמציג הודעת שגיאה.
- ניתן להשתמש במאפיין Visibleכדי להסתיר או להציג רכיבי ממשק משתמש.
- רכיב ה- NxtDriveיכול להניע, לסובב ולעצור את הרובוט.
- ניתן להשתמש ברכיב NxtUltrasonicSensorכדי לזהות מכשולים.

אמזון בחנות הספרים

איור 13-1.

נניח שאתה מעיין בספרים בחנות הספרים האהובה עליך ורוצה לדעת כמה א עלויות הזמנה באתר Amazon.com. עם האמזון ב את אפליקציית חנות הספרים, אתה יכול לסרוק ספר או להזין ISBN, והאפליקציה תגיד לך את המחיר הנמוך ביותר הנוכחי של הספר באתר Amazon.com. אתה יכול גם לחפש ספרים בנושא מסוים.

אמזון בחנות הספרים מדגים איך [illegible] תוכדי ליצור אפליקציות שמדברות לשירותי אינטרנט (המכונה ממשקי תכנות יישומים או ממשקי API). אפליקציה זו תקבל נתונים משירות אינטרנט שנוצר על ידי אחד ממחברי הספר הזה. עד סוף פרק זה, תוכל ליצור אפליקציה מותאמת אישית משלך לשיחה עם אמזון. לאפליקציה יש ממשק משתמש פשוט שבאמצעותו המשתמש יכול להזין מילות מפתח או ISBN של ספר (מספר ספר סטנדרטי בינלאומי - קוד בן 10 או 13 ספרות המזהה ספר באופן ייחודי) ולאחר מכן מפרט את הכותרת, ISBN והמחיר הנמוך ביותר עבור עותק חדש באמזון. הוא גם משתמש ברכיב BarcodeScanner כך שהמשתמש יכול לסרוק ספר כדי להפעיל חיפוש במקום להזין טקסט (טכנית, הסורק פשוט מזין עבורך את ה-NBSI של הספר).

מה תלמד

באפליקציה זו (מוצגת באיור ,(1-13תלמדו:

•כיצד להשתמש בסורק ברקוד בתוך

אפליקציה.

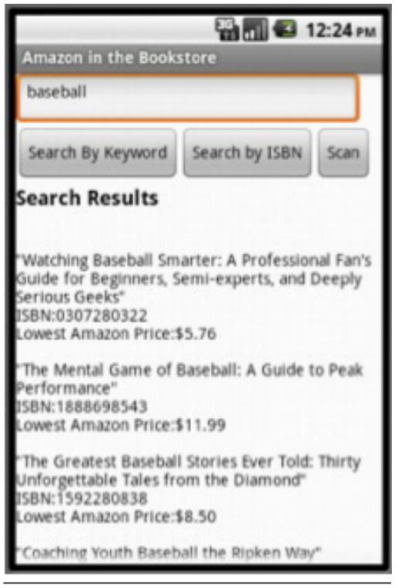
•כיצד לגשת למקור מידע אינטרנטי

API-ה)של אמזון) דרך ה- TinyWebDB

רְכִיב.

•כיצד לעבד נתונים מורכבים המוחזרים מאותו מקור מידע אינטרנטי. בפרט, תלמד כיצד לעבד רשימה של ספרים שבה

כל ספר הוא עצמו רשימה של שלושה פריטים (כותרת, מחיר ו-NBSI).



איור .13-2אמזון בחנות הספרים
פועלת באמולטור

תוצג לך גם קוד מקור שבו תוכל להשתמש כדי ליצור אינטרנט משלך
שירות APIעם שפת התכנות Python ppA-וEngine של גוגל.

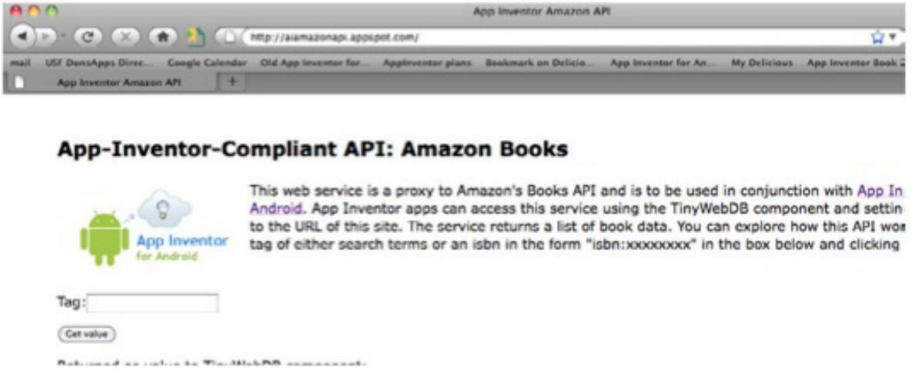
מהו API?

לפני שתתחיל לעצב את הרכיבים שלך ולתכנת את האפליקציה, בואו נסתכל מקרוב על מה זה ממשק מתכנת יישומים (API)וכיצד הוא פועל. APIהוא כמו אתר אינטרנט, אבל במקום לתקשר עם בני אדם, הוא מתקשר עם תוכנות מחשב אחרות. ממשקי APIנקראים לעתים קרובות תוכניות "שרת" מכיוון שהם בדרך כלל מגישים מידע לתוכניות "לקוח" שמתממשקות למעשה עם בני אדם -כמו אפליקציית .App Inventorאם אי פעם השתמשת באפליקציית Facebook בטלפון שלך, אתה משתמש בתוכנת לקוח שמתקשרת עם אפליקציית שרת API-השל Facebook.
בפרק זה, תיצור אפליקציית לקוח אנדרואיד המתקשרת עם ממשק APIשל אמזון.
האפליקציה שלך תבקש מידע על ספר ו-NBSI ,Amazon API-מוה-IPA יחזיר רישומים מעודכנים לאפליקציה שלך. לאחר מכן האפליקציה תציג את נתוני הספר למשתמש.

API-השל Amazonשבו תשתמש מוגדר במיוחד לשימוש עם .App Inventor אֲנִי
לא יכנס כאן לפרטים העגומים, אבל כדאי לדעת שכתוצאה מהקונפגורציה הזו, אתה יכול להשתמש ברכיב TinyWebDBכדי לתקשר עם אמזון.
החדשות הטובות הן שאתה כבר יודע איך לעשות את זה! אתה תתקשר ל- TinyWebDB.GetValue כדי לבקש מידע ולאחר מכן תעבד את המידע המוחזר במטפל האירועים , TinyWebDB.GotValue בדיוק כפי שאתה עושה כשאתה משתמש במסד נתונים אינטרנטי. (תוכל לחזור ליישומי MakeQuizו-TaQekaiu בפרק 10כדי לרענן את הזיכרון שלך, במידת הצורך.)

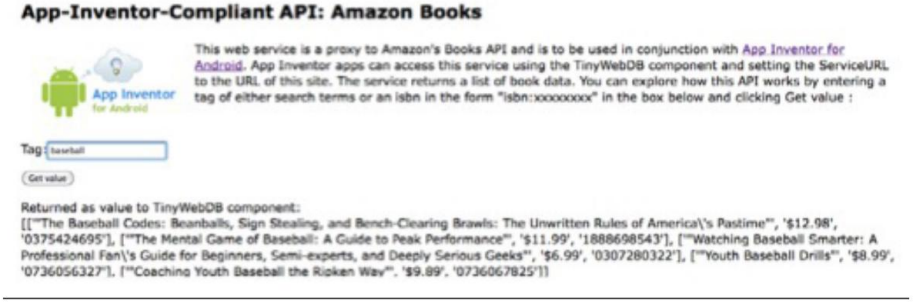
לפני יצירת האפליקציה, תצטרך להבין את הפרוטוקול של ,Amazon APIהמפרט את הפורמט עבור הבקשה שלך ואת הפורמט של הנתונים המוחזרים. בדיוק כפי שלתרבויות אנושיות שונות יש פרוטוקולים שונים (כשאת פוגשת מישהו, אתה לוחץ ידיים, מרכין או מהנהן?), גם למחשבים שמדברים זה עם זה יש פרוטוקולים. API-השל אמזון שבו תשתמש כאן מספק ממשק אינטרנט לבחינת אופן הפעולה של API-הלפני שתתחיל להשתמש בו. למרות שה-IPA נועד לדבר עם מחשבים אחרים, ממשק אינטרנט זה מאפשר לך לראות איך התקשורת הזו תתרחש. בעקבות השלבים הללו, תוכלו לנסות אילו קריאות GetValueמסוימות יחזרו דרך האתר, ולדעת שממשק API-היתנהג בדיוק אותו הדבר כאשר תבקשו ממנו נתונים דרך רכיב TinyWebDB ב-ppA Inventor.בואו נתחיל:

1.פתח דפדפן ועבור אל http://aiamazonapi.appspot.com/.אתה תראה את אתר המוצג באיור 13-2.



איור .13-3ממשק האינטרנט של ה-ppA Inventor Amazon API

2.בדף אינטרנט זה, אתה יכול לנסות את הפונקציה האחת שאתה יכול לקרוא עם API-ההזה:
.getvalueהזן מונח (למשל, "בייסבול") בשדה התגים ולאחר מכן לחץ על "קבל ערך".
דף האינטרנט יציג רשימה של חמשת הספרים המובילים שהוחזרו מאמזון, כפי שמוצג באיור .13-3

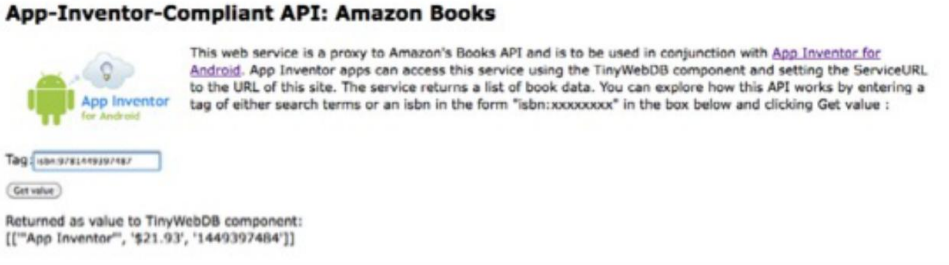


איור .13-4קריאה Amazon API-לכדי לחפש ספרים הקשורים לתג (או מילת המפתח) "בייסבול"

הערך המוחזר הוא רשימה של ספרים, כל אחד מוקף בסוגריים [ככה]
ומספק את הכותרת, העלות וה-NBSI.
אם תסתכל מקרוב, תראה שכל ספר מיוצג למעשה כרשימה משנה
של רשימה ראשית אחרת. הרשימה הראשית (על בייסבול) מוקפת בסוגריים, וכל תת-רשימה (או ספר) מוקפת בקבוצת סוגריים משלה בתוך הסוגריים הראשיים. אז, ערך ההחזרה API-מזה הוא למעשה רשימה של רשימות, כאשר כל תת-רשימה מספקת את המידע עבור ספר אחד. בואו נסתכל על זה קצת יותר מקרוב. כל סוגר שמאלי ([)בנתונים מציין את תחילתה של רשימה. הסוגר השמאלי הראשון של התוצאה מציין את תחילת הרשימה החיצונית (רשימת הספרים). מימין לה ההתחלה של רשימת המשנה הראשונה, הספר הראשון, כפי שמוצג כאן:

["קודי הבייסבול: כדורי כדורים, גניבת שלטים וקטטות ספסל:
החוקים הבלתי כתובים של הבילוי של אמריקה," ['0375424695' ',$12.98'
לרשימה המשנה שלושה חלקים: שם, המחיר הנוכחי הנמוך ביותר עבור הספר באמזון
ו-NBSI של הספר. כאשר אתה מקבל מידע זה לאפליקציית App Inventorשלך, תוכל לגשת
לכל חלק באמצעות פריט רשימה נבחר, עם אינדקס 1עבור הכותרת, אינדקס 2עבור המחיר
ואינדקס 3עבור ה-NBSI. (כדי לרענן את הזיכרון בעבודה עם אינדקס ורשימות, בקר שוב
באפליקציית MakeQuizבפרק (.10

.3במקום לחפש לפי מילת מפתח, ניתן לחפש ספר על ידי הזנת ISBN. כדי לבצע חיפוש כזה, אתה מזין תג בצורת "isbn: 00000000000"כאשר רשימת ה-0 מייצגת מספר ISBNבפועל (ראה איור .(4-13הסוגריים הכפולים ([])בתוצאה [["'ממציא אפליקציות,'" [['1449397484' ',$21.93'מציינים שרשימה של רשימות עדיין מוחזרת, למרות שיש רק ספר אחד. זה אולי נראה קצת מוזר עכשיו, אבל זה יהיה חשוב כשאנחנו ניגשים למידע עבור האפליקציה שלנו.



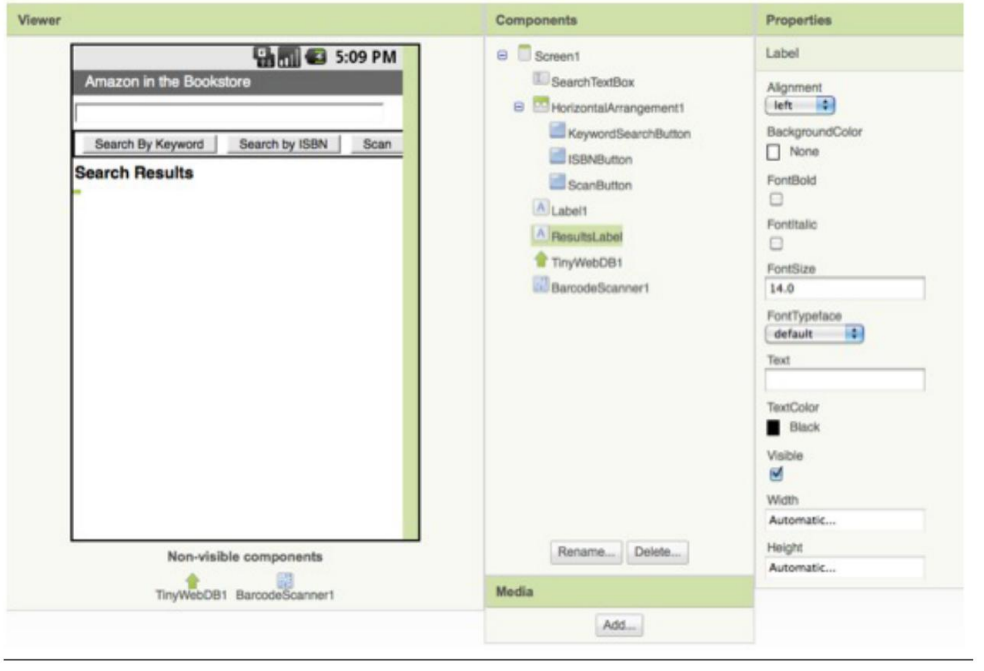
איור .13-5חיפוש API-השל אמזון באמצעות ISBNבמקום מילת מפתח

מתחילים

התחבר לאתר App Inventorוהתחל פרויקט חדש. אז תן לזה "AmazonBooks",והגדר את כותרת המסך "Amazon at the Bookstore"-ללאחר מכן, חבר את המכשיר או האמולטור שלך לבדיקה חיה.

עיצוב הרכיבים

ממשק המשתמש של אפליקציית הספרים של אמזון הוא פשוט יחסית: תן לה Textboxלהזנת מילות מפתח או ISBN, שני כפתורים להפעלת שני סוגי החיפושים (מילת מפתח או ISBN),וכפתור שלישי לאפשר למשתמש לסרוק ספר (אנחנו נגיע לזה עוד מעט). לאחר מכן, הוסף תווית כותרת ותווית נוספת לרשימת התוצאות שה-nozamA API מחזיר, ולבסוף שני רכיבים שאינם גלויים: TinyWebDB ו- .BarcodeScannerבדוק את התוצאות שלך מול איור .13-5



איור .13-6ממשק המשתמש של אמזון בחנות הספרים המוצג במעצב

טבלה 13-1מפרטת את כל הרכיבים שתצטרך לבניית ממשק המשתמש המוצג באיור .13-5

טבלה .13-1רשימת רכיבים עבור אמזון באפליקציית חנות הספרים

מַטָרָה	קבוצת צבעים איך תקרא לזה		סוג רכיב
המשתמש מזין מילות מפתח או ISBN כאן.	ממשק משתמש SearchTextBox		תיבת טקסט
[illegible]	[illegible]		[illegible]
ממשק משתמש [illegible]	[illegible]		לַחְצָן
ממשק משתמש [illegible] ISBN	[illegible]		לַחְצָן
לחץ [illegible] ISBN [illegible]	[illegible]		לַחְצָן
הכותרת [illegible] 1	[illegible]		תווית
היכן [illegible]	[illegible]		תווית
דבר עם Amazon.com [illegible]	[illegible]		TinyWebDB
סורק ברקוד [illegible]	[illegible]		סורק ברקוד

הגדר את המאפיינים של הרכיבים בצורה הבאה:

1.הגדר את הרמז של SearchTextBoxל"הזן מילות מפתח או ISBN".

.2הגדר את המאפיינים של הכפתורים והתוויות כך שהם יופיעו כפי שמוצג ב
איור .5-13

.3הגדר את המאפיין ServiceURL של רכיב TinyWebDB ל-
. http://aiamazonapi.appspot.com/

תכנות התנהגות האפליקציה

עבור אפליקציה זו, תציין את ההתנהגויות הבאות בעורך הבלוקים:

חיפוש לפי מילת מפתח

המשתמש מזין כמה מונחים ולוחץ על כפתור חיפוש מילות מפתח כדי להפעיל חיפוש באמזון. אתה תתקשר ל- TinyWebDB1.GetValue כדי שזה יקרה.

חיפוש לפי ISBN

המשתמש מזין ISBN ולוחץ על כפתור ISBNB. אתה תארז את הקידומת "isbn:" עם המספר שהוזן ותפעיל את החיפוש של אמזון.

סריקת ברקוד

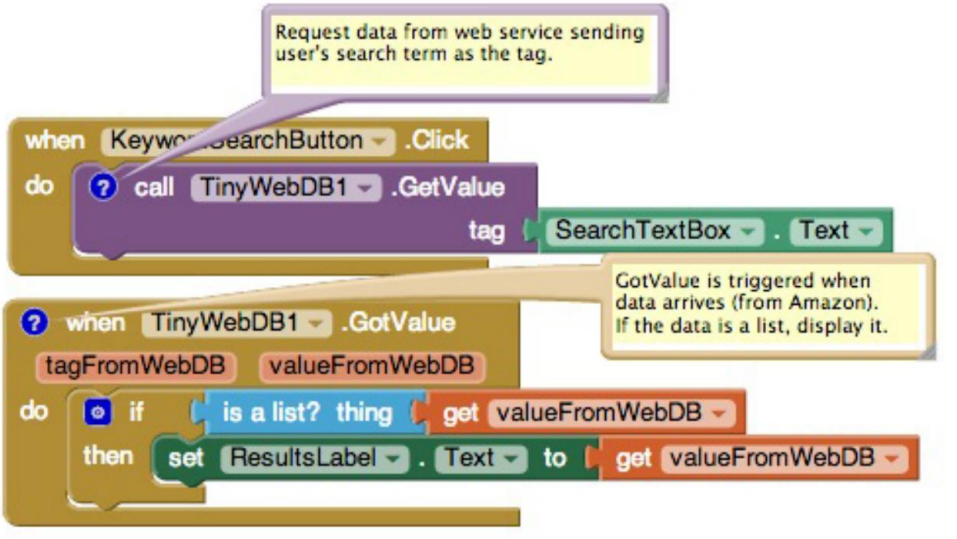
המשתמש לוחץ על כפתור והסורק מופעל. כאשר המשתמש סורק ISBN מתוך ספר, האפליקציה שלך תפעיל חיפוש לפי ISBN.

עיבוד רשימת הספרים

בהתחלה, האפליקציה שלך תציג את הנתונים שהוחזרו מאמזון בצורה בסיסית. מאוחר יותר, תשנה את הבלוקים כך שהאפליקציה תחלץ את הכותרת, המחיר וה-NBSI מכל ספר שהוחזר ותציג אותם בצורה מסודרת.

חיפוש לפי מילת מפתח

כאשר המשתמש לוחץ על כפתור KeywordSearch, אתה רוצה לתפוס את הטקסט מ-SearchTextBox ולשלוח אותו בתור התג בבקשה שלך לתשתמש בגוש Amazon API-. TinyWebDB.GetValue כדי לבקש את החיפוש באמזון. כאשר התוצאות יחזרו מאמזון, מטפל האירועים TinyWebDB.GotValue יופעל. לעת עתה, פשוט הצג את התוצאה שמוחזרת ישירות לתוויות התוצאות, כפי שמוצג באיור 13-6. בהמשך, לאחר שתראה שהנתונים אכן מאוחזרים, תוכל להציג את הנתונים בצורה מתוחכמת יותר.



איור 13-7. שלח את בקשת החיפוש API-לושם תוצאות ResultsLabel-ב

איך הבלוקים עובדים

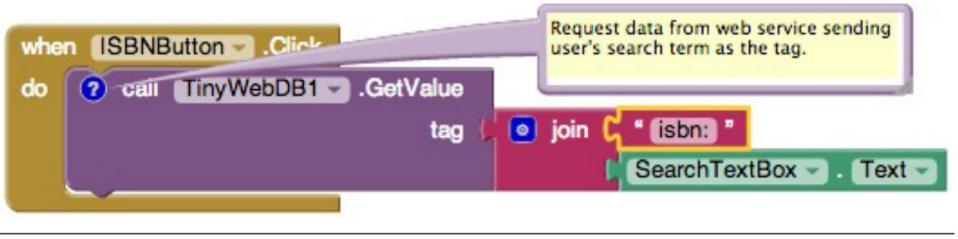
כאשר המשתמש לוחץ על כפתור KeywordSearch, מתבצעת בקשת TinyWebDB1.GetValue. התג שנשלח עם הבקשה הוא המידע שהמשתמש הזין ב-SearchTextBox. אם השלמת את אפליקציית MakeQuiz (פרק 10), אתה יודע שבקשות TinyWebDB1.GetValue לא נענות מיד. במקום זאת, כאשר הנתונים מגיעים מה-IPA, TinyWebDB1.GotValue מופעל. ב- GotValue, הבלוקים בודקים את הערך המוחזר כדי לראות אם מדובר ברשימה (זה לא יהיה אם API-השל Amazon אינו מקוון או שאין נתונים עבור מילות המפתח). אם מדובר ברשימה, הנתונים ממוקמים ב- ResultsLabel.

בדוק את האפליקציה שלך הזן מונח בתיבת החיפוש ולחץ על חפש לפי מילת מפתח. אתה אמור לקבל רשימה דומה למה שמוצג באיור 13-7. (זה לא נורא יפה למראה, אבל נטפל בזה בקרוב.)

איור 13-8. תוצאת חיפוש של מילות מפתח עבור "כלבים"

חיפוש לפי ISBN

הקוד לחיפוש לפי ISBN דומה, אבל במקרה זה API-השל Amazon מצפה שהתג יהיה בצורה "isbn:xxxxxxxxxxxxx"הפרוטוקול שה-IPA מצפה לחיפוש לפי .(ISBN אתה לא רוצה ISBN) והז) לאלץ את המשתמש לדעת את הפרוטוקול הזה; המשתמש צריך רק להיות מסוגל להזין את ה-NBSI בתיבת הטקסט וללחוץ על חפש לפי ISBN, והאפליקציה צריכה להוסיף את הקידומת "isbn:"מאחורי הקלעים עם טקסט עשה. איור 13-8 מציג את הבלוקים לעשות זאת.



איור 13-9. האפליקציה מקדימה "isbn:"לחיפוש כך שהיא תחפש ספר מסוים

איך הבלוקים עובדים

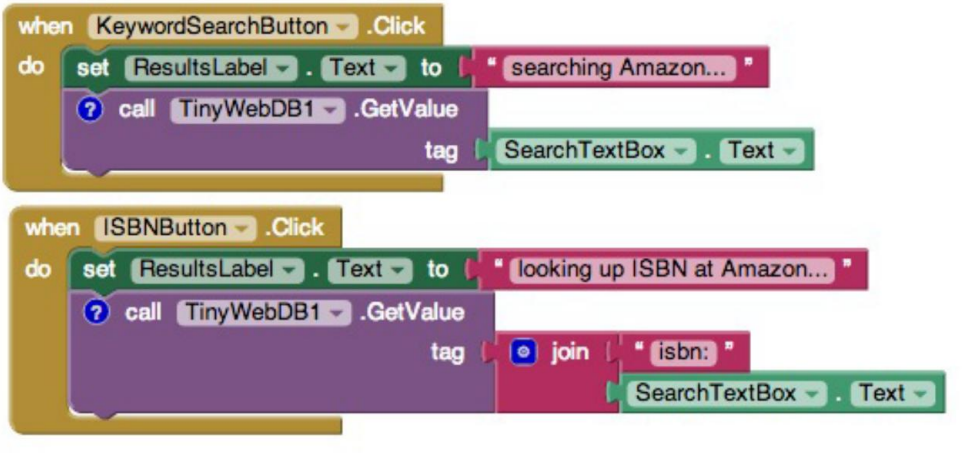
בלוק הצטרפות משרשר את הקידומת "isbn:"עם המידע שהמשתמש הזין ב- SearchTextBox ושולח את התוצאה כתג ל- TinyWebDB1.GetValue.

בדיוק כמו בחיפוש מילות מפתח, API-השולח חזרה תוצאת רשימה עבור חיפוש ISBN-in במקרה זה, רשימה של הפריט היחיד שה-NBSI שלו תואם בדיוק את הקלט של המשתמש. מכיוון שמטפל האירועים TinyWebDB.GotValueכבר מוגדר לעבד רשימה של ספרים (אפילו רשימה עם פריט אחד בלבד), לא תצטרך לשנות את מטפל האירועים שלך כדי שזה יעבוד.

בדוק את האפליקציה שלך הזן ISBN(למשל, 9781449397487)ולחץ על האם פרטי הספר מופיעים? ISBNB מתאר SearchTextBox

אל תשאיר את המשתמשים שלך תלויים

כאשר אתה מתקשר לשירות אינטרנט (API)עם TinyWebDB1.GetValue, יכול להיות עיכוב לפני הגעת הנתונים ו- TinyWebDB1.GotValueמופעל. בדרך כלל מומלץ ליידע את המשתמשים שהבקשה מעובדת כדי להרגיע אותם שהאפליקציה לא נתקעה. עבור אפליקציה זו, אתה יכול לשים הודעה ב- ResultsLabelבכל פעם שאתה מבצע את הקריאה ל- TinyWebDB1.GetValue, כפי שמוצג באיור 13-9.



איור 13-10. הוספת הודעה כדי ליידע את המשתמש מה קורה

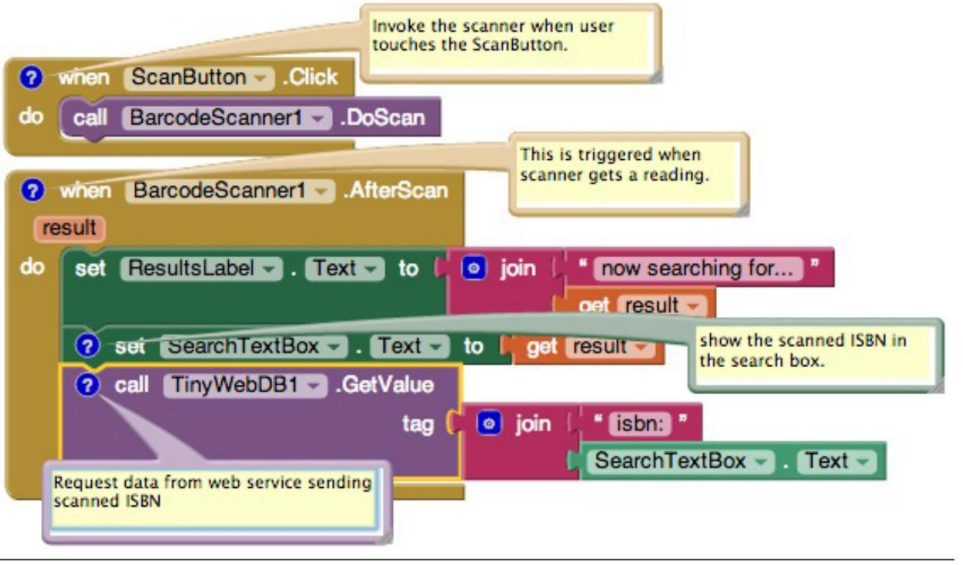
איך הבלוקים עובדים

הן עבור מילת המפתח והן עבור חיפושי ISBN, הודעת "חיפוש באמזון..." ממוקמת ב- ResultsLabel כאשר הנתונים מתבקשים. שימו לב שכאשר GotValueמופעלת, הודעה זו נמחקת עם התוצאות בפועל מאמזון.

סריקת ספר

בואו נודה בזה: הקלדה בפלאפון היא לא תמיד הדבר הכי קל, ואתם נוטים לטעות פה ושם. זה בהחלט יהיה קל יותר (ויגרום לפחות טעויות) אם משתמש יוכל פשוט להפעיל את האפליקציה שלך ולסרוק את הברקוד של הספר. זוהי עוד תכונת טלפון אנדרואיד מובנית נהדרת שתוכל לנצל אותה בקלות עם App Inventor.

הפונקציה BarcodeScanner.DoScan מפעילה את הסורק. אתה תרצה לקרוא לזה כאשר לוחצים על כפתור הסריקה . המטפל באירועים BarcodeScanner.AfterScan הוא מופעל ברגע שמשהו נסרק. יש לו ארגומנט אחד, תוצאה, שמכיל את המידע שנסרק. במקרה זה, אתה רוצה להתחיל חיפוש ISBN באמצעות תוצאה זו, כפי שמוצג באיור 13-10.



איור 13-11. חסימות לתחילת חיפוש ISBN לאחר סריקה של משתמש

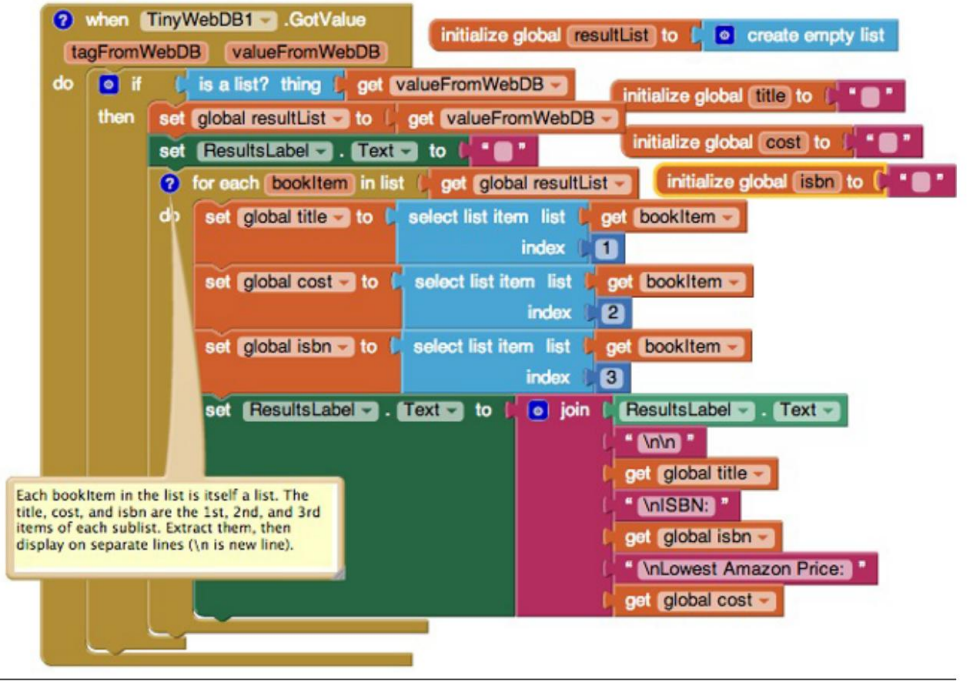
איך הבלוקים עובדים

כאשר המשתמש לוחץ על לחצן הסריקה , DoScan מפעיל את הסורק. כאשר משהו נסרק, AfterScan מופעל. תוצאת הטיעון מכילה את תוצאת הסריקה -במקרה זה, ISBN של ספר. המשתמש מקבל הודעה על כך שנעשתה בקשה, התוצאה (מספר ה-NBSI הסרוק) ממוקמת ב- ,SearchTextBox ו- TinyWebDB1.GetValue נקרא להתחיל את החיפוש. שוב, מטפל האירועים TinyWebDB1.GotValue יעבד את פרטי הספר המוחזרים.

בדוק את האפליקציה שלך לחץ על לחצן הסריקה וסרוק את הברקוד של ספר. האם האפליקציה מציגה את המידע על הספר?

שיפור התצוגה

אפליקציית לקוח כמו זו שאתה יוצר יכולה לעשות מה שהיא רוצה עם הנתונים שהיא מקבלת -אתה יכול להשוות את פרטי המחיר לזה של חנויות מקוונות אחרות, או להשתמש בפרטי הכותרת כדי לחפש ספרים דומים מספרייה אחרת. כמעט תמיד, תרצה לטעון את מידע API-הלמשתנים שאותם תוכל לעבד הלאה. במטפל האירועים TinyWebDB.GotValue שיש לך עד כה, אתה פשוט מכניס את כל המידע שהוחזר מאמזון ל- .ResultsLabelבמקום זאת, בואו נעבד את הנתונים על ידי (1הוספת הכותרת, המחיר וה-NBSI של כל ספר שהוחזר למשתנים נפרדים, ו-2) הצגת הפריטים הללו בצורה מסודרת. אם השלמת חלק מהפרקים הקודמים, כנראה שאתה מבין בהגדרת משתנים ושימוש בהם בתצוגה שלך, אז נסה לבנות את המשתנים שאתה חושב שתזדקק להם ואת הבלוקים להצגת כל תוצאת חיפוש בה. קו נפרד משלו. לאחר מכן, השווה את מה שעשית עם איור 13-11.



איור 13-12. חילוץ הכותרת, העלות וה-NBSI של כל ספר, ולאחר מכן הצגתם בשורות נפרדות

איך הבלוקים עובדים

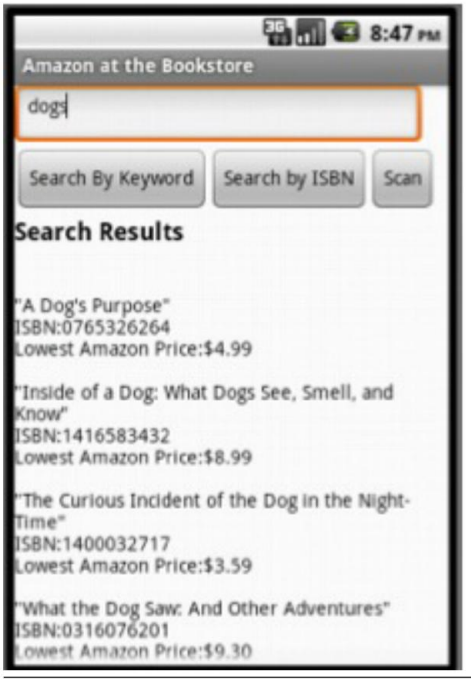
ארבעה משתנים, ResultList - כותרת, עלות ו- isbn - מוגדרים כך שיכילו כל פיסת נתונים כפי שהיא מוחזרת מהממשק API. ההתוצאה מה-IPA, valueFromWebDB ממוקמת במשתנה resultList. אפליקציה זו הייתה יכולה לעבד את הארגומנט valueFromWebDB ישירות, אבל באופן כללי, תכניס אותו למשתנה למקרה שתרצה לעבד את הנתונים מחוץ למטפל באירועים. (ארגומנטים של אירוע כמו valueFromWebDB מחזיקים את הערך שלהם רק בתוך המטפל באירועים.)

עבור כל לולאה משמש כדי לחזור על כל פריט של התוצאה. נזכיר כי ה A הנתונים המוחזרים מאמזון הם רשימה של רשימות, כאשר כל תת-רשימה מייצגת את המידע עבור ספר. אז, שם placeholder- השל עבור כל אחד שונה לפריט ספר, והוא מכיל את פרטי הספר הנוכחיים (רשימה) בכל איטרציה.

כעת עלינו להתמודד עם העובדה שהפריט המשתנה הוא רשימה - הפריט הראשון היא הכותרת; השני, המחיר; והשלישי, ה-NBSI. לפיכך, אנו משתמשים בלוקים של פריטי בחירה כדי לחלץ את הפריטים הללו ולהציב אותם במשתנים המתאימים (כותרת, מחיר ו- isbn).

לאחר שהנתונים אורגנו למשתנים, תוכל לעבד אותם איך שתרצה. האפליקציה הזו פשוט משתמשת במשתנים כחלק מבלוק הצטרפות שמציג את הכותרת, המחיר וה-NBSI בשורות נפרדות.

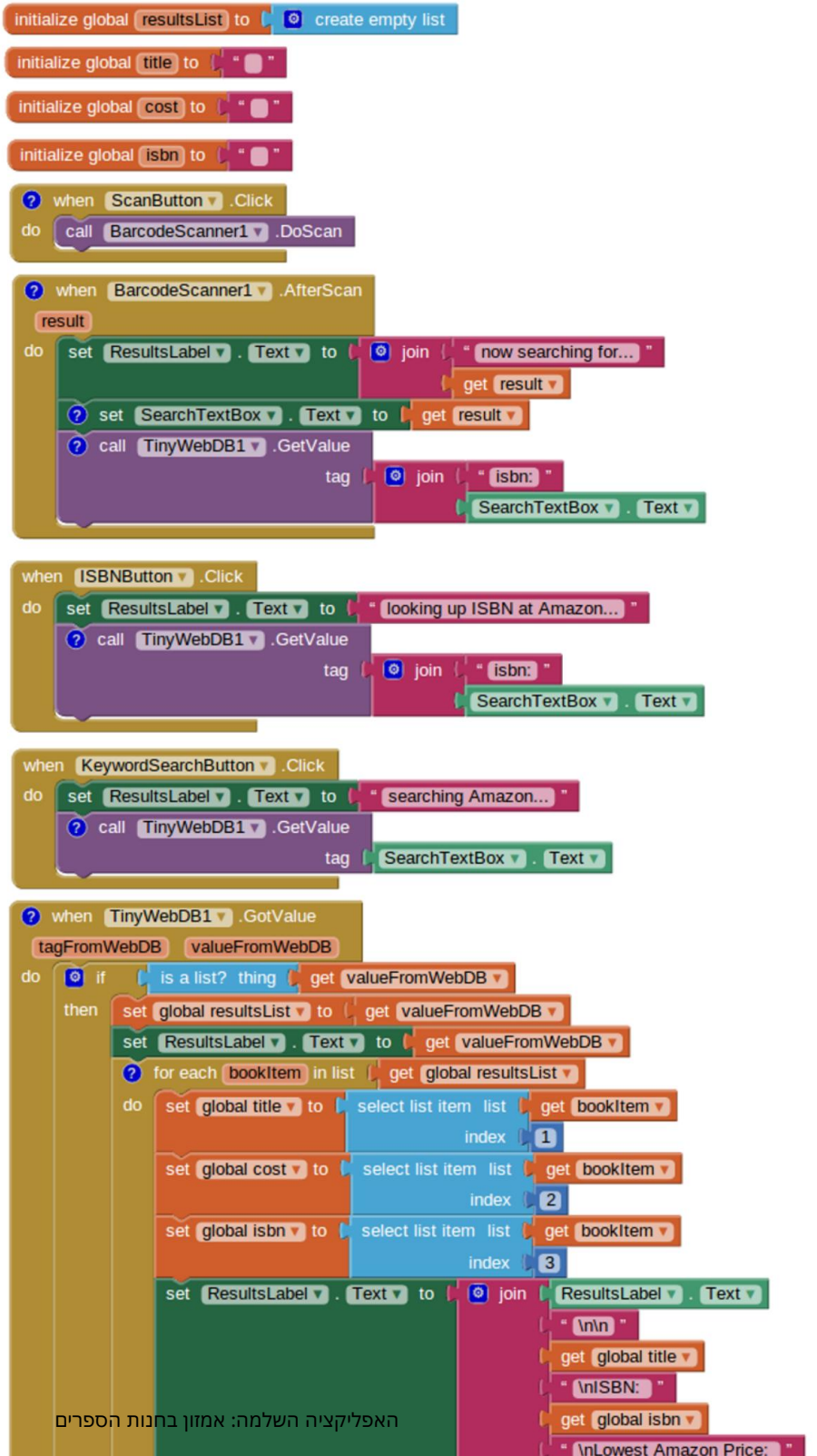
בדוק את האפליקציה שלך נסה חיפוש נוסף ובדוק כיצד המידע על הספר מוצג. זה צריך להיראות דומה לתמונה 12-13.



איור 13-13. רישום החיפוש הוצג בצורה מתוחכמת יותר

האפליקציה השלמה: אמזון בחנות הספרים

איור 13-13 מציג את תצורת הבלוק הסופי עבור אמזון בחנות הספרים.



התאמה אישית של ה-API

נוצר עם שפת התכנות Python ו-App Engine של גוגל. App Engine מאפשר לך ליצור ולפרוס אתרים ושירותים החיים בשרתים של Google. אתה משלם עבור App Engine רק אם האתר או ה-API שלך הופכים פופולריים ומושכים הרבה כניסות.

ה-API שהמשמש במדריך זה מספק גישה חלקית בלבד ל-Amazon API להמלא ומחזיר לכל היותר חמישה ספרים לכל חיפוש. אם תרצה לספק גמישות רבה יותר - לדוגמה, בקש ממנו לחפש פריטים שאינם ספרים - תוכל להוריד את קוד המקור של השירות מ- http://appinventorapi.com/ ולהתאים אותו אישית. התאמה אישית כזו אכן דורשת ידע בתכנות Python, אז היזהרו! אבל, אם סיימתם את אפליקציות ה-App Inventor בספר זה, אולי אתם פשוט מוכנים לאתגר. כדי להתחיל ללמוד Python, בדוק את הגרסה האינטראקטיבית של הספר How to Think Like a Computer Scientist: Learning with Python ולאחר מכן עיין בסעיף על בניית ה-API של App Inventor בפרק 24 של ספר זה.

וריאציות

לאחר שתפעיל את האפליקציה, אולי תרצה לחקור כמה מהווריאציות הבאות:

- כפי שהיא, האפליקציה נתקעת אם החיפוש לא מחזיר ספרים (לדוגמה, כאשר המשתמש מזין ISBN לא חוקי). שנה את החסימות כך שהאפליקציה תדווח כשאין תוצאות.
- שנה את האפליקציה כך שתציג רק ספרים מתחת ל-$01.
- שנה את האפליקציה כך שלאחר סריקת ספר, המחיר הנמוך ביותר באמזון יושמע בקול (השתמש ברכיב TextToSpeech שנדון באפליקציית No Text While Driving בפרק 4).
- הורד את http://aiamazonapi.appspot.com קוד API ושנה אותו כך שיחזיר מידע נוסף. לדוגמה, ייתכן שתבקש ממנו להחזיר את כתובת האתר של אמזון של כל ספר, להציג את כתובת האתר יחד עם כל ספר ברשימה, ולתת למשתמש ללחוץ על כתובת האתר כדי לפתוח את הדף הזה. כפי שהוזכר קודם לכן, שינוי ה-API הדורש תכנות של Python וידע מסוים ב-App Engine של גוגל.

למידע נוסף, עיין בפרק 24.

סיכום

הנה כמה מהמושגים שכיסינו באפליקציה הזו:

•אתה יכול לגשת לאינטרנט מאפליקציה באמצעות TinyWebDBובמיוחד
ממשקי APIשנבנו. אתה מגדיר את ה- ServiceURLשל רכיב TinyWebDBלכתובת API-הולאחר מכן
מתקשר ל- TinyWebDB.GetValueכדי לבקש את המידע. הנתונים אינם מוחזרים מיד, אלא ניתן לגשת
אליהם בתוך המטפל באירועים . TinyWebDB.GotValue

•הפונקציה BarcodeScanner.DoScanמפעילה את הסריקה. כאשר המשתמש סורק את א
ברקוד, האירוע BarcodeScanner.AfterScanמופעל והנתונים הסרוקים ממוקמים בתוצאת הארגומנט.

•ב-Inventor, ppAנתונים מורכבים מיוצגים עם רשימות ורשימות של רשימות. אם אתה יודע את הפורמט של
הנתונים המוחזרים ,API-מאתה יכול להשתמש עבור כל פריט רשימה ובחר כדי לחלץ את פיסות המידע
הנפרדות למשתנים, ולאחר מכן לבצע כל עיבוד או להגדיר את התצוגה איך שתרצה באמצעות המשתנים האלה .