

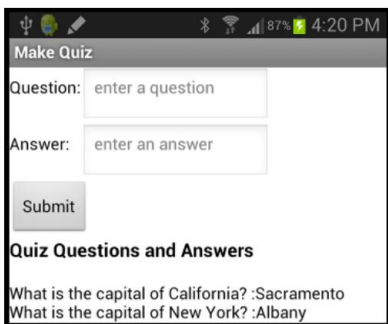
MakeQuiz I-TakeQuiz

איור 10-1.



אתה יכול להתאים אישית את אפליקציית חידון הנשיאים בפרק 8 כדי לבנות כל חידון, אבל זה רק המתכנת שיכול 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 (!App Inventor). בפרק זה, תבנה אפליקציית MakeQuiz המאפשרת "מורה" יוצר חידונים באמצעות טופס קלט. השאלות והתשובות של החידון יאוחסנו במסד נתונים אינטרנטי כך ש"סטודנטים" יוכלו לגשת לאפליקציית TakeQuiz נפרדת ולגשת למבחן. תוך כדי בניית שתי האפליקציות הללו, תבצע עוד קפיצת מדרגה קונספטואלית משמעותית: למד כיצד ליצור אפליקציות עם נתונים שנוצרו על ידי משתמשים המשותפים בין אפליקציות ו

משתמשים.



איור 10-2. אפליקציית MakeQuiz בפעולה

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

תעבד שתי אפליקציות, MakeQuiz עבור המורה (ראה איור I-TakeQuiz ו-10-1 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, TakeQuiz תעבוד בדומה לאפליקציית Presidents Quiz שכבר בנית. למעשה, תשתמש באפליקציית חידון הנשיאים כנקודת התחלה. TakeQuiz יהיה 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 MakeQuiz I-TakeQuiz תלמדו כיצד לבנות אפליקציה שמטפלת בנתונים משותפים שנוצרו על ידי משתמשים.

אם השלמת את אפליקציית Xylophone (פרק 9), כבר הוצגת בפניך לרשימות דינמיות; באפליקציה זו, הנתונים המוזיקליים שהמשתמש מגן 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 שסופקו על ידי משתמש הקצה. מדרך זה מכסה את המושגים הבאים של ממציא האפליקציות:

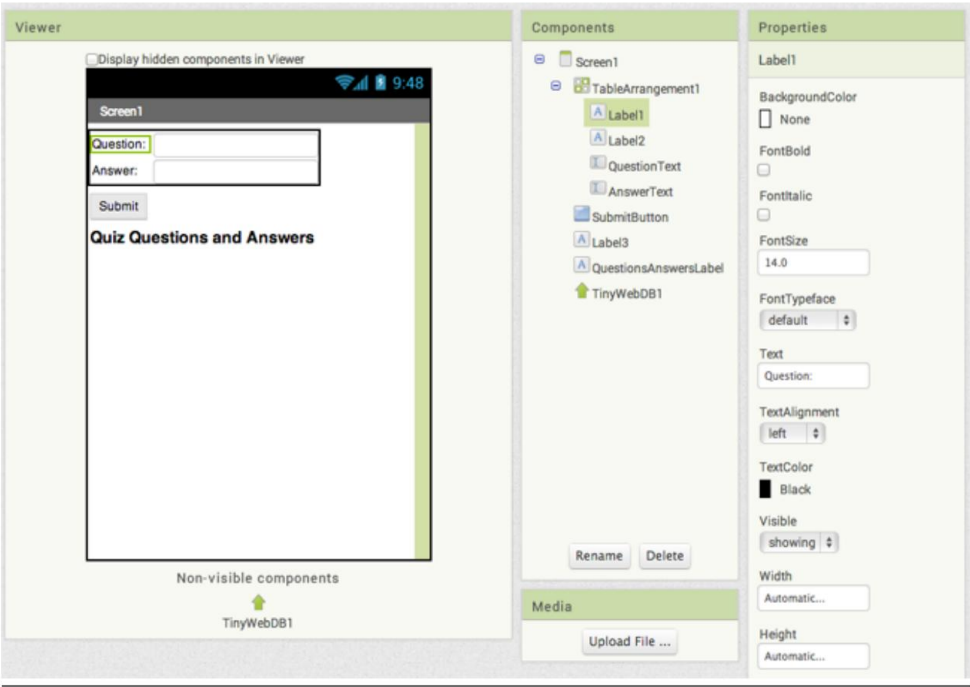
- טפסי קלט לאפשר למשתמש להזין מידע.
- שימוש ברשימה באינדקס יחד עם עבור כל אחד מהם כדי להציג פריטים מרשימות מרובות.
- נתוני רשימה מתמשכת MakeQuiz - ישמור את השאלות והתשובות של החידון במסד נתונים אינטרנטי, TakeQuiz ויטען אותן מאותו מסד נתונים.
- שיתוף נתונים - אתה תשמור את הנתונים במסד נתונים אינטרנטי באמצעות רכיב TinyWebDB (במקום רכיב TinyDB ששימש בפרקים הקודמים).

מתחילים

התחבר לאתר App Inventor והתחל 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 MakeQuiz. כשאתה 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 10-2 על ידי גרירת ה-רכיבים המפורטים בטבלה 10-1. גרור כל רכיב מהלוח לתוך ה-מציג ושם אותו כפי שצוין בטבלה. שים לב שאתה יכול להשאיר את תוויות הכותרת שמות (Label1 - Label4) כברירת המחדל שלהם (לא תשתמש בהם בעורך הבלוקים בכל מקרה).

טבלה 10-1. כל הרכיבים לאפליקציית MakeQuiz

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

170פרק 10: MakeQuiz ו-TakeQuiz

מְטָרָה	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	לחצן
הצג את "שאלות ותשובות חידון".	תווית ממשק משתמש3	תווית
שאלות ממשק משתמש תשובות תווית	הצגת צמדי QAשהוזנו בעבר.	תווית
אחסון 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 Label3ל"שאלות ותשובות חידון".

2.הגדר את FontSize של Label3 ל-81 וסמן את התיבה FontBold.

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

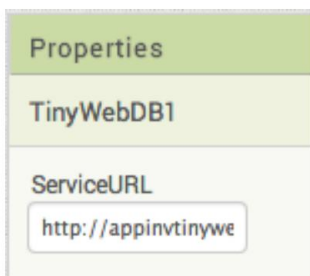
4.הגדר את הטקסט של כפתור "Submit" ל-Submit.

5.הגדר את הטקסט של QuestionsAnswersLabel ל"שאלות ותשובות חידון".

6.העבר את QuestionText, AnswerText והתוויות המשויות אליהם לתוך סידור השולחן 1.

אם תסתכל על המאפיינים של TinyWebDB, תבחין שיש לו מאפיין ServiceURL (ראה איור 3-10) מאפיין זה מציין שירות מסד נתונים אינטרנטי, במיוחד מוגדר לעבוד עם רכיב TinyWebDB, היכן הנתונים המשותפים שלך יהיו מאוחסן. כברירת מחדל, שירות האינטרנט שאליו הוא מתייחס הוא כזה שהוגדר על ידי ממציא האפליקציות של MIT צוות בכתובת http://appinvtinywebdb.appspot.com.אחרי שתתחבר לשירות ברירת המחדל הזה בזה הדרכה תוך כדי עבודה; עם זאת, חשוב לדעת שכל מי שמשתמש ב-App Inventor 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, שתמצא 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-TinyWebDB and "ממשקי API ותואמי TinyWebDB" בעמוד 368.



איור 4-10. 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 4-10.



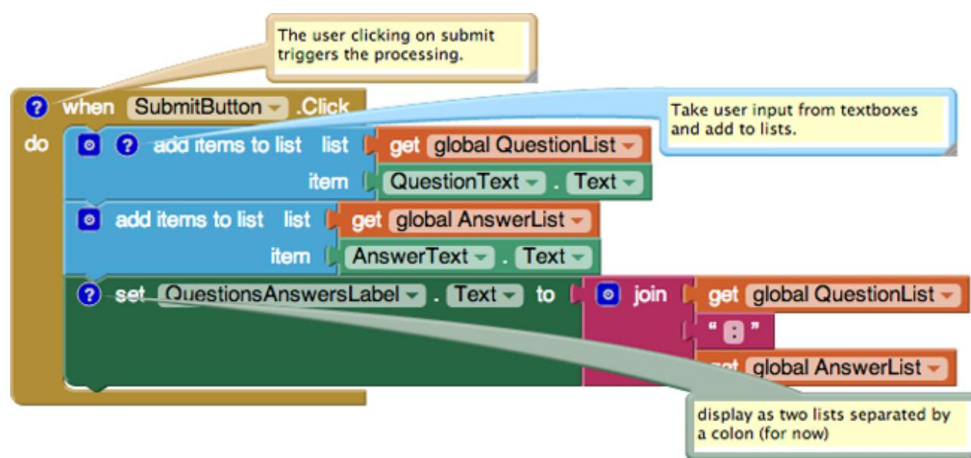
איור 5-10. הרשימות של MakeQuiz מתחילות ריקות

הרשימות מוגדרות באמצעות בלוק צור רשימה ריקה, במקום בלוק הפוך לרשימה. הסיבה לכך היא שעם האפליקציות TakeQuiz ו-MakeQuiz 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 השאלות והתשובות. הבלוקים צריכים להופיע כמתואר באיור 5-10.

פרק 172 I-TakeQuiz ו-MakeQuiz



איור 10-6. הוספת ערכים חדשים לרשימות

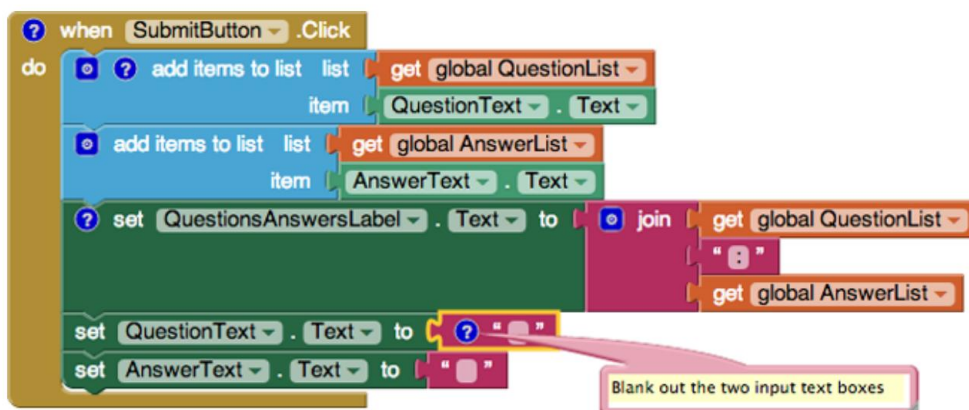
איך הבלוקים עובדים

בלוק הוסף פריטים לרשימה מוסיף כל פריט לסוף רשימה. כפי שמוצג באיור 10-5, האפליקציה לוקחת את הטקסט שהשתמש הזין בתיבות הטקסט `QuestionText` ו-`AnswerText` ומצרפת כל אחת מהן לרשימה המתאימה.

הוספת הפריטים לרשימה בלוקים מעדכנים את המשתנים `QuestionList` ו-`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, "(פריט 1 פריט 2)". 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 שמציגות כל צמד תשובות לשאלה בשורה נפרדת.

מחיקת השאלה והתשובה

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



איור 7-10. ביטול 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- QuestionText ו- AnswerText מרוקן עם 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. 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 אחד בכל שורה, כך:

מהי בירת קליפורניה? סקרמנטו מהי בירת ניו יורק? אלבני הטכניקה להצגת רשימה בודדת עם כל פריט בשורה נפרדת מתוארת בפרק 20 - כדאי 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

10: MakeQuiz I-TakeQuiz פרק 174

בשם, `displayQAs` וקראו לנוהל הזה מהמטפל באירועים `SubmitButton.Click`.

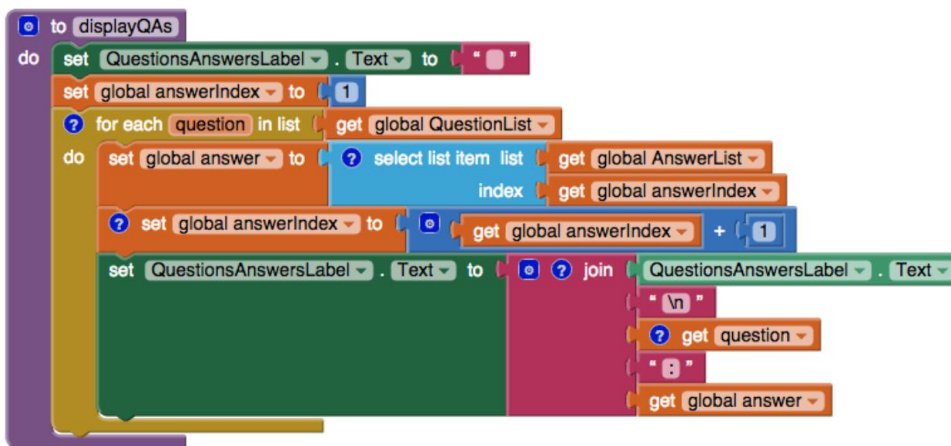
כדי להציג צמדי שאלות-תשובות בשורות נפרדות, יהיה עליך לבצע את הפעולות הבאות:

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

• השתמש ב-`answerIndex` משתנה כדי שתוכל לתפוס כל תשובה בזמן שאתה חוזר על השאלות.

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

הבלוקים צריכים להופיע כמתואר באיור 10-7.



איור 10-8. נוהל `displayQAs`

איך הבלוקים עובדים

הליך `displayQAs` מקפל את כל הבלוקים להצגת הנתונים.

על ידי שימוש בנוהל, לא תצטרך להעתיק את הבלוקים הדרושים להצגת הרשימה יותר מפעם אחת באפליקציה -אתה יכול פשוט להתקשר `displayQAs`-לכאשר אתה צריך להציג את הרשימות.

עבור כל אחד בלבד מאפשר לך לחזור על רשימה בודדת. במקרה הזה, יש שתי רשימות, `QuestionList` ו-`AnswerList`. הערך 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-`currentQuestionIndex` במדריך חידון הנשיאים בפרק 8. במקרה זה, משתנה האינדקס, `answerIndex` משמש כדי לעקוב אחר המיקום ב-`AnswerList` כאשר עבור כל אחד מהם עובר רשימת השאלות.

answerIndex מוגדר ל-1 לפני תחילת עבור כל אחד. בתוך עבור כל אחד, answerIndex משמש לבחירת התשובה הנוכחית מרשימת התשובות, ולאחר מכן היא מוגדלת. בכל איטרציה של עבור כל אחד, השאלה והתשובה הנוכחיות משורשרות לסוף המאפיין, QuestionsAnswersLabel.Text עם נקודתיים ביניהן.

התקשרות לנוהל התצוגה

כעת יש לך נוהל להצגת צמדי השאלות והתשובות, אבל זה לא יעזור אלא אם תתקשר אליו כשתצטרך. שנה את המטפל באירוע SubmitButton.Click על ידי קריאה ל-displayQAs במקום להציג את הרשימות, כפי שנעשה בעבר. הבלוקים המעודכנים צריכים להופיע כפי שמוצג באיור 10-8.



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



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

שמירת QAS-הבהתמדה 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 המכונת שלי? אפליקציה (פרק 7), אתה יודע שאם לא תשמור את הנתונים במסד נתונים, הם לא יהיו שם כשהשתמש יוצא ומפעיל מחדש את האפליקציה. אחסון הנתונים באופן מתמיד יאפשר ליצור חידון להציג או לערוך את העדכון האחרון של החידון בכל פעם שהאפליקציה מופעלת. אחסון מתמשך נחוץ גם מכיוון שאפליקציית TakeQuiz זקוקה גם לגישה לנתונים. אתה כבר מכיר את השימוש ברכיב TinyDB לאחסון ואחזור נתונים במסד נתונים. אבל במקרה זה, במקום זאת, תשתמש ברכיב TinyWebDB. TinyWebDB מאחסנת מידע ישירות בטלפון, TinyWebDB מאחסנת 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

10: MakeQuiz I-TakeQuiz פרק 176

שניהם זקוקים לגישה לאותם נתונים - אם יצרנית החידון מאחסנת את השאלות והתשובות בטלפון שלה, למתמודדי החידון לא תהיה כל דרך להגיע לנתונים עבור החידון שלהם! מכיוון ש- TinyWebDB מאחסנת 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:

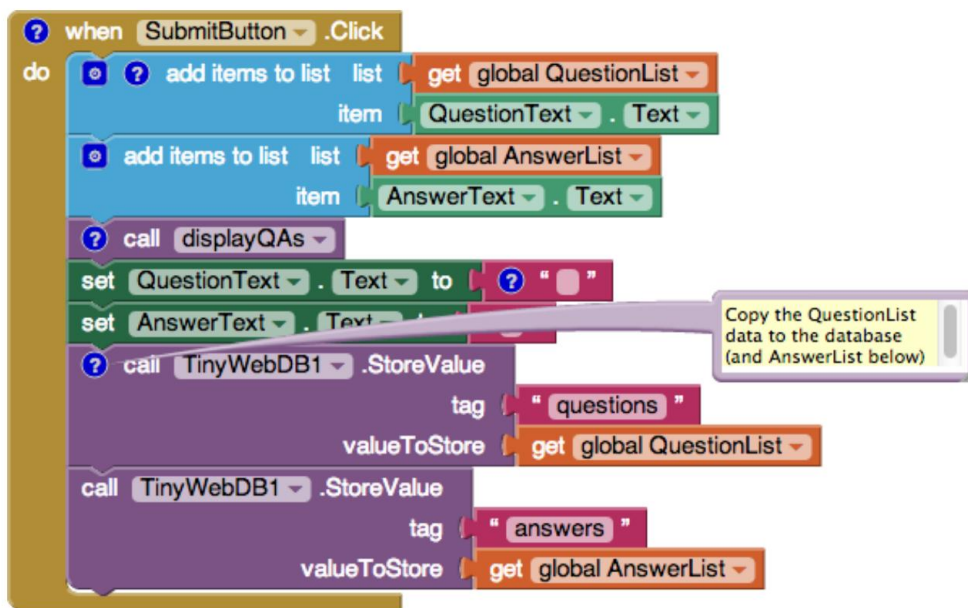
• אחסון רשימה במסד הנתונים בכל פעם שמתווסף לו פריט חדש.

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



איור 10-10. אחסון השאלות והתשובות במאגר

עבור האפליקציה שלך, עליך להשתמש בתגים ייחודיים יותר מ"שאלות" ו"תשובות" (למשל, "DavesQuestions" ו-"DavesAnswers" 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.



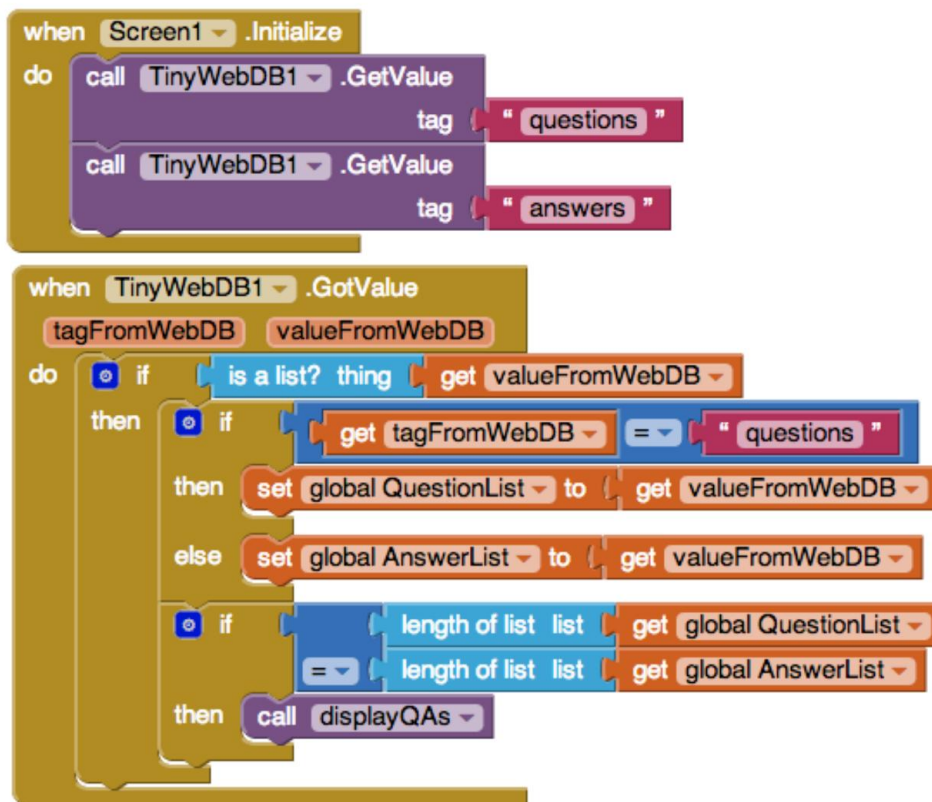
בדוק את האפליקציה שלך בדיקת חלק זה של האפליקציה שונה מבדיקות שביצעת בעבר מכיוון שהאפליקציה שלך משפיעה כעת על ישות אחרת, שירות ברירת המחדל של TinyDBWeb. הפעל את האפליקציה, הזן שאלה ותשובה ולאחר מכן פתח חלון דפדפן לשירות האינטרנט המוגדר כברירת מחדל בכתובת <http://appinvtinywebdb.appspot.com> לאחר מכן לחץ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

כפי שצוין 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 API ותואמי TinyWebDB ו-TinyWebDB 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 באירועים . 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 הבולקים עובדים

בולקים 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; זה לא מקבל מיד ערך. במקום זאת, כאשר הנתונים מגיעים ממסד הנתונים האינטרנטי, מופעל אירוע TinyWebDB.GotValue. עליו גם לתכנת את אותו מטפל באירועים לעבד את הנתונים המוחזרים.

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

באפליקציה הזו, מכיוון שמבוצעות שתי בקשות שונות עבור השאלות והתשובות, GotValue יופעל פעמיים. כדי להימנע מהוספת 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 תשובות רשימת).

ב-, Screen.Initialize האפליקציה קוראת ל- TinyWebDB1.GetValue פעמיים: פעם אחת כדי לבקש את השאלות המאוחסנות, ופעם אחת כדי לבקש את ה- AnswerList המאוחסנת. כאשר הנתונים מגיעים ממסד הנתונים באינטרנט מכל אחת מהבקשות, האירוע TinyWebDB1.GotValue מופעל.

הארגומנט valueFromWebDB של GotValue מחזיק את הנתונים המוחזרים מבקשת מסד הנתונים. אתה צריך את החסימה החיצונית אם במטפל האירועים כי מסד הנתונים יחזיר טקסט ריק ("") ב- valueFromWebDB. הפעם הראשונה שבה נעשה שימוש באפליקציה ועדיין אין שאלות ותשובות. אם תשאל אם ה- valueFromWebDB הוא 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- tagFromWebDB: יהיה "שאלות" או "תשובות". אם התג הוא "שאלות", ה- valueFromWebDB מוכנס למשתנה QuestionList אחרת (אחר), הוא ממוקם ברשימת התשובות. (אם השתמשת בתגים שאינם "שאלות" ו"תשובות", בדוק את אלה, במקום זאת).

אתה רוצה להציג את הרשימות רק לאחר ששתייהן הגיעו (GotValue) הופעל פעמיים). האם אתה יכול לחשוב כיצד תוכל לדעת בוודאות ששתי הרשימות נטענות ממסד הנתונים? הבלוקים המוצגים משתמשים במבחן אם כדי לבדוק אם

אורכי הרשימות זהים, מכיוון שזה יכול להיות נכון רק אם שתי הרשימות הוחזרו. אם כן, הליך displayQAs השימושי שכתבת קודם נקרא כדי להציג את הנתונים הטעונים.

האפליקציה השלמה: MakeQuiz

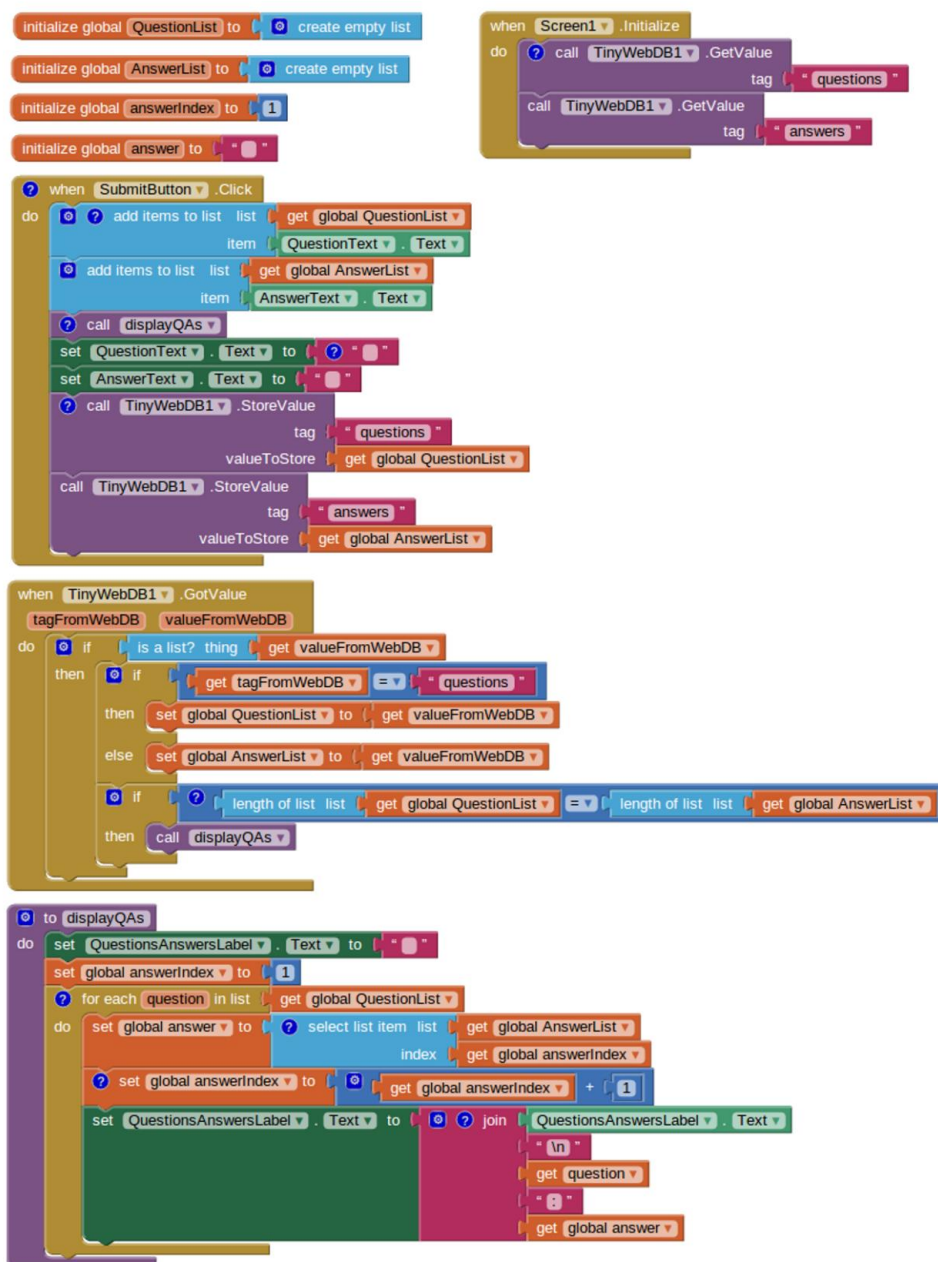
אורך 10-11 מציג את הבלוקים עבור כל אפליקציית MakeQuiz.

TakeQuiz: אפליקציה לביצוע החידון ב- מאגר מידע

כעת יש לך אפליקציית MakeQuiz שתאחסן 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 Quiz שלך, App Inventor-בבחירה באפשרות Save As, שם הפרויקט החדש. "TakeQuiz" זה ישאיר את אפליקציית Presidents Quiz שלך ללא שינוי, אבל עכשיו אתה יכול להשתמש בלוקים שלה כבסיס ל-TakeQuiz.

פרק 180: MakeQuiz ו-TakeQuiz



אזור 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 TakeQuiz. b-Component Designer, TakeQuiz

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

2.מכיוון TakeQuiz-שיעבוד עם נתוני מסד נתונים שנמצאים ברשת, גרור רכיב TinyWebDB לתוך האפליקציה.

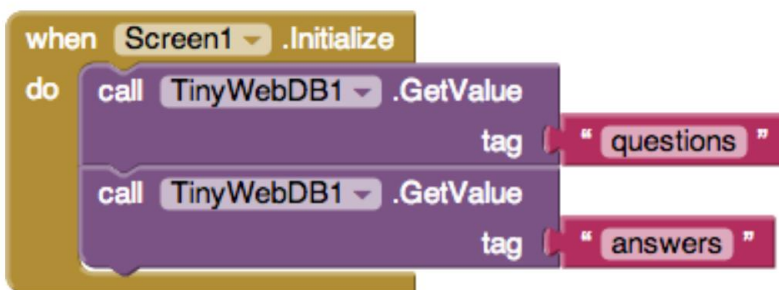
3.מכיוון שאינך רוצה שהמשתמש יענה או ילחץ על NextButton עד שהשאלות ייטענו, בטל את הסימון של המאפיין Enabled של ה- AnswerButton -
כפתור הבא.

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 QuestionList - AnswerList. הבלוקים המתקבלים צריכים להופיע כפי שמוצג באיור 10-12.



איור 10-13. רשימות השאלות והתשובות מתחילות כעת ריקות

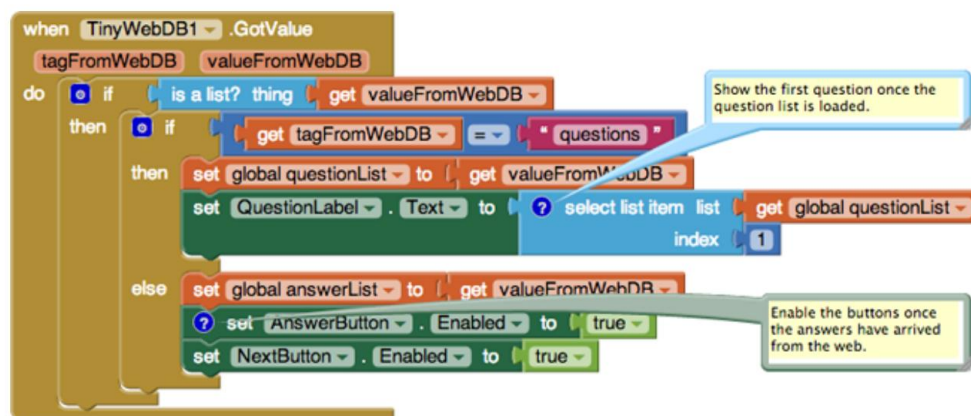
אתה יכול גם למחוק לחלוטין את PictureList; האפליקציה הזו לא תעסוק בתמונות. כעת, שנה את Screen1.Initialize שלך כך שיקרא ל- TinyWebDB.GetValue פעמיים כדי לטעון את הרשימות, בדיוק כפי שעשית MakeQuiz-בהבלוקים צריכים להיראות כפי שהם נראים באיור 10-13.



איור 10-14. בקשת השאלות והתשובות ממאגר האינטרנט

לבסוף, גרור החוצה מטפל באירועים TinyWebDB.GetValue . מטפל באירועים זה אמור להיראות דומה לזה המשמש MakeQuiz-באך כאן אתה רוצה להציג רק את השאלה הראשונה ואף אחת מהתשובות. נסה לבצע את השינויים האלה בעצמך תחילה, ולאחר מכן עיין בבלוקים באיור 10-14 כדי לראות אם הם תואמים את הפתרון שלך.

10: MakeQuiz I-TakeQuiz פרק 182



איור 10-15. GetValue בתונים המגיעים מהרשת

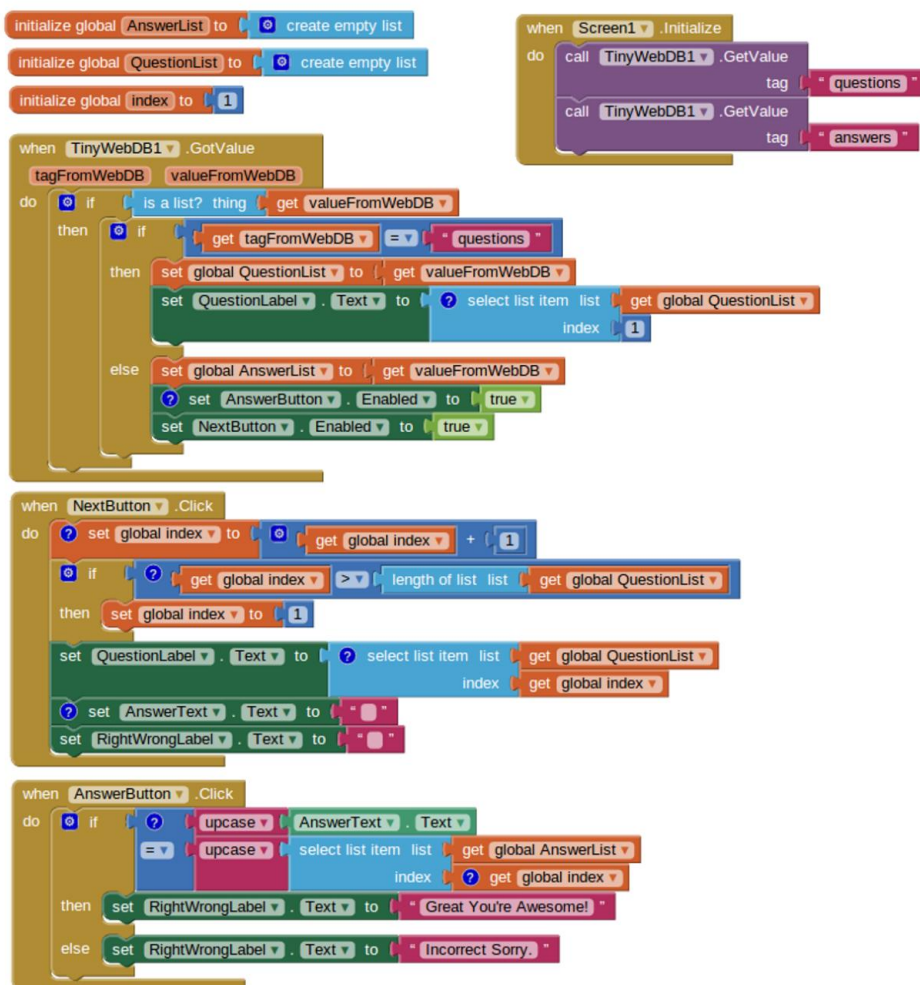
איך הבולקים עובדים

כאשר האפליקציה מתחילה, Screen1.Initialize מופעל והאפליקציה מבקשת את השאלות והתשובות ממסד הנתונים האינטרנטי. כאשר כל בקשה מגיעה, המטפל באירוע TinyWebDB.GetValue מופעל. האפליקציה בודקת תחילה אם אכן יש נתונים ב-valueFromWebDB באמצעות האם רשימה? אם היא מוצאת נתונים, האפליקציה שואלת איזו בקשה נכנסה, באמצעות tagFromWebDB וממקמת את ה-valueFromWebDB ברשימה המתאימה. אם השאלות נטען, השאלה הראשונה נבחרת מ-QuestionList ומוצגת. אם AnswerList-הנטען, ה-AnswerButton ו-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 הבולקים עבור כל אפליקציית TakeQuiz.



איור 16-10 החסימות הסופיות עבור TakeQuiz

וריאציות

לאחר שתגרום i-TakeQuiz ו-MakeQuiz ללעבוד, אולי תרצה לחקור כמה מהוריאציות הבאות:

- אפשר ליוצר החידון לציין תמונה עבור כל שאלה. זה קצת מסובך כי TinyWebDB לא מאפשר לך לאחסן תמונות. לכן, התמונות יצטרכו להיות כתובות URL לתמונות באינטרנט, ויוצר החידון יצטרך להזין כתובות URL אלו כפריט שלישי בטופס MakeQuiz. שים לב שאתה יכול להגדיר את המאפיין Picture של רכיב תמונה לכתובת URL.

• אפשר ליוצר החידון למחוק פריטים מהשאלות והתשובות. אתה יכול לתת למשתמש לבחור שאלה באמצעות רכיב `ListPicker` ותוכל להסיר פריט עם בלוק הסר פריט רשימה (זכור להסיר משתי הרשימות ולעדכן את מסד הנתונים). לעזרה עם `ListPicker` ומחיקת רשימה, ראה פרק 19.

• תן ליצרן החידון לתת שם לחידון. תצטרך לאחסן את שם החידון תחת תג אחר במסד הנתונים, ותצטרך לטעון את השם יחד עם החידון `TakeQuiz`-בלאחר שטענת את השם, השתמש בו כדי להגדיר את המאפיין `Screen.Title` כך שיופיע כאשר המשתמש ניגש לחידון.

• אפשר ליצרן 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` רכיב.

• אתה מאחזר נתונים ממסד נתונים של `TinyWebDB` על ידי בקשתם באמצעות `TinyWebDB.GetValue` כאשר מסד הנתונים באינטרנט מחזיר את הנתונים, האירוע `TinyWebDB.GotValue` מופעל. במטפל האירועים, `TinyWebDB.GotValue` אתה יכול לשים את הנתונים ברשימה או לעבד אותם בדרך כלשהי.

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