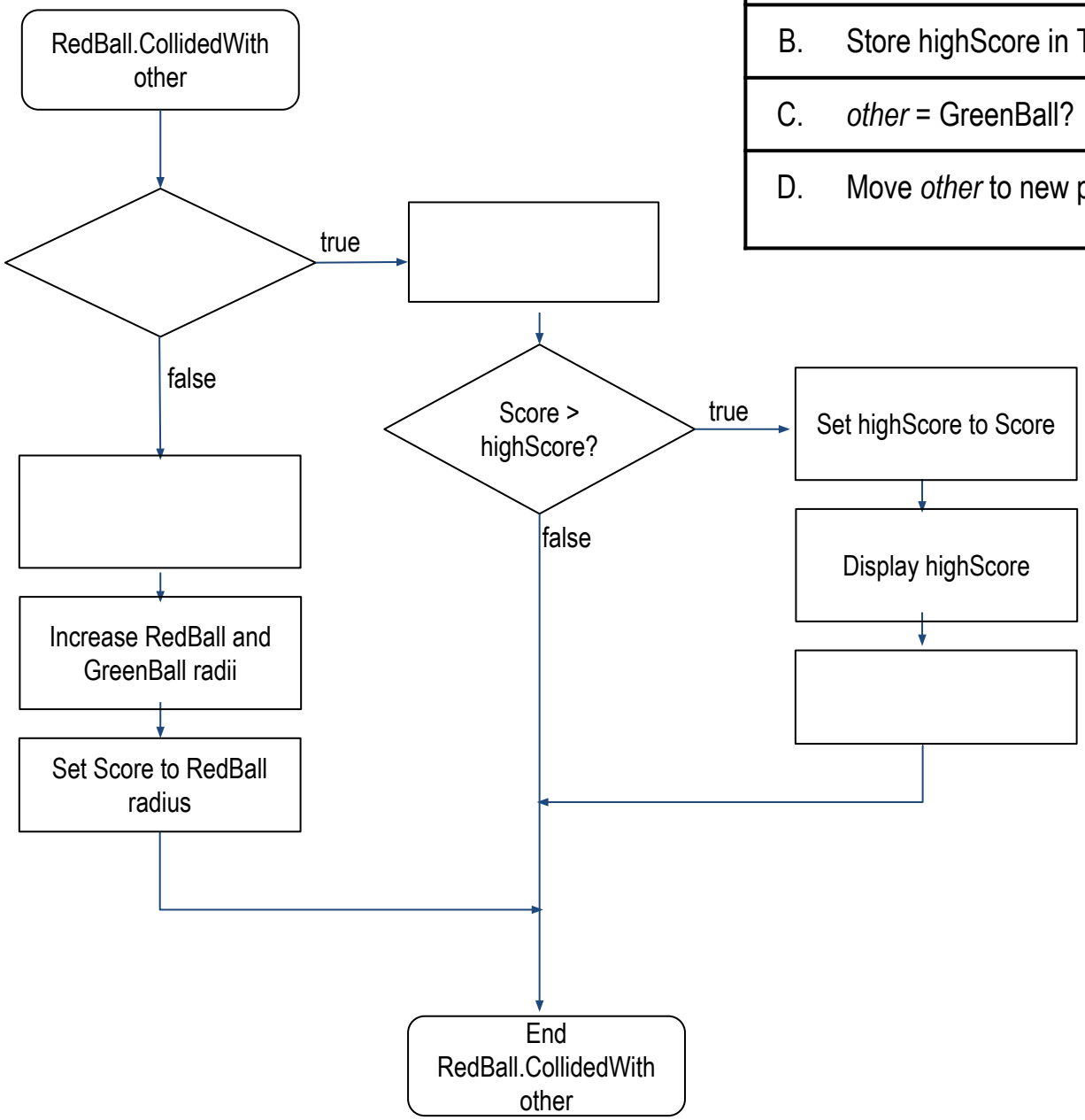
FOOD CHASE GAME: PART 4

You will add a High Score for the game, using the final radius of the RedBall as a "score". The bigger the better, right?

START HERE

1 With your partner, look at the following flowchart and fill in the missing blocks with the correct letter, according to this table.

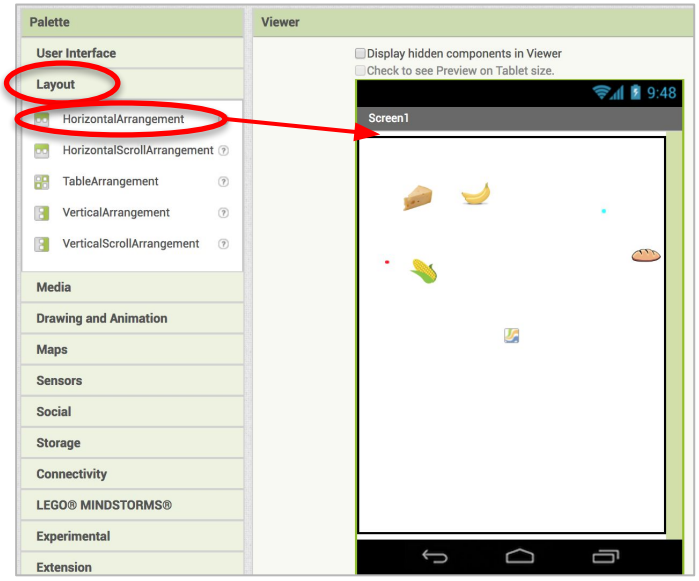
A.	Notify user game is over
B.	Store highScore in TinyDB
C.	*other* = GreenBall?
D.	Move *other* to new position



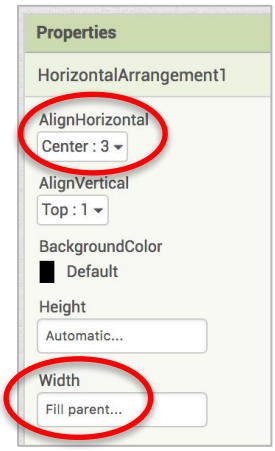
ADDING COMPONENTS

2 Go to the Designer.

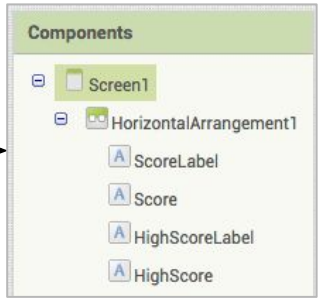
3 At the top of the Viewer, add a **HorizontalArrangement**.

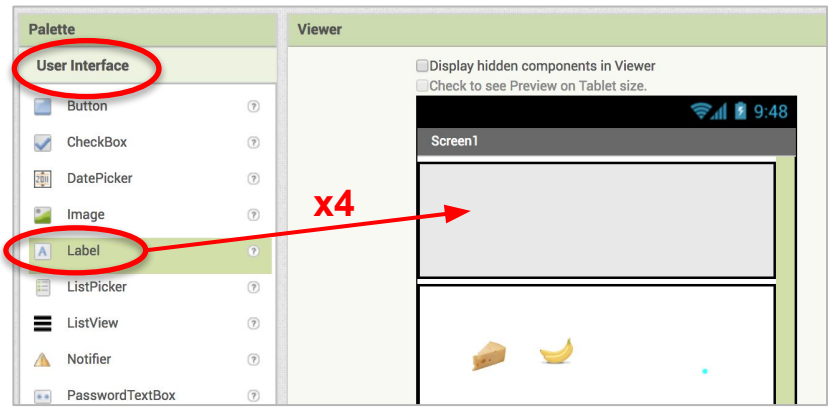


4 Set its *Width* to "**Fill Parent**". and its *AlignHorizontal* to "**Center**".



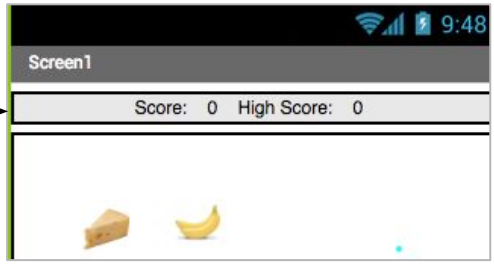
5 Drag four **Labels** into the **HorizontalArrangement**. Name them in order as shown below.





6 Set the *Text* for each **Label** as follows:

- **ScoreLabel**: "**Score:** "
- **Score**: "**0**"
- **HighScoreLabel**: "**High Score:** "
- **HighScore**: "**0**"

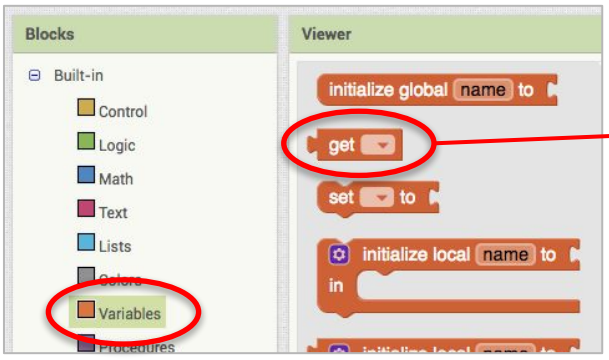


VARIABLES

7 From the Variables drawer, drag out a new **initalize global name** block and change the name to “highScore”.

8 Initialize it to zero - **0**.

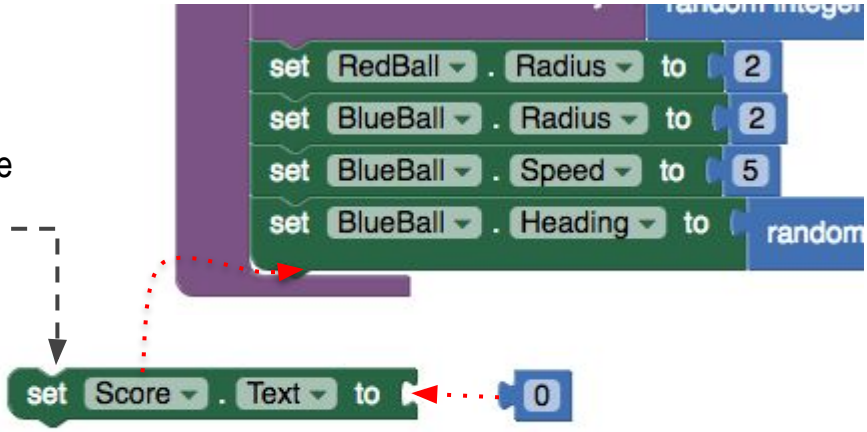
9 In the **Screen1.Initialize** event block, add blocks to set the Label **HighScore**.**Text** to the value of the variable.



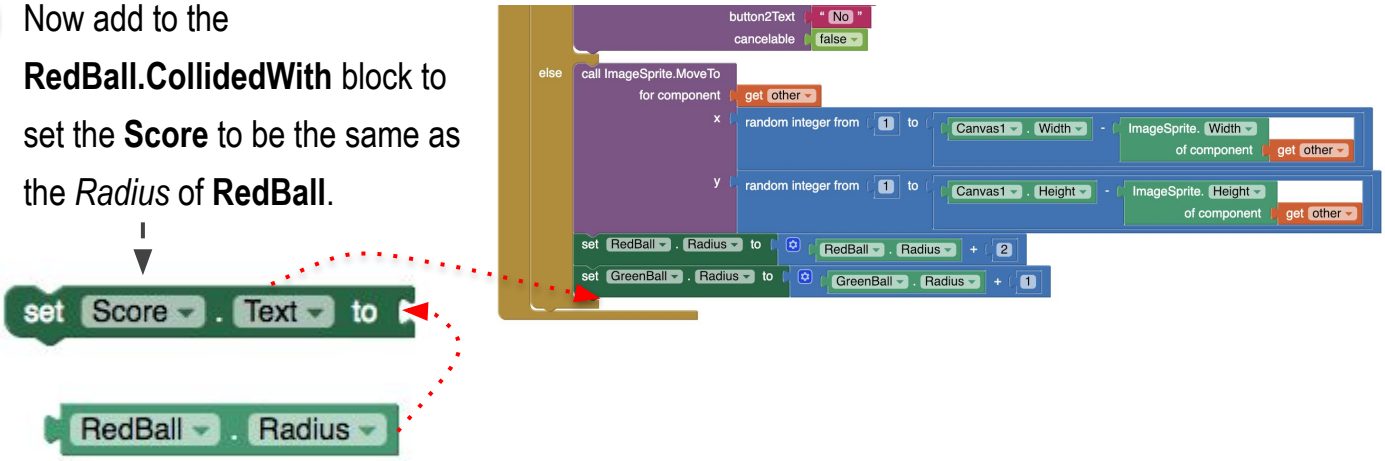
The **get** block *gets* the value of the variable. Use the dropdown menu to select highScore.

SETTING THE SCORE

10 Set the **Score** Label back to **0** in the **Restart** procedure.

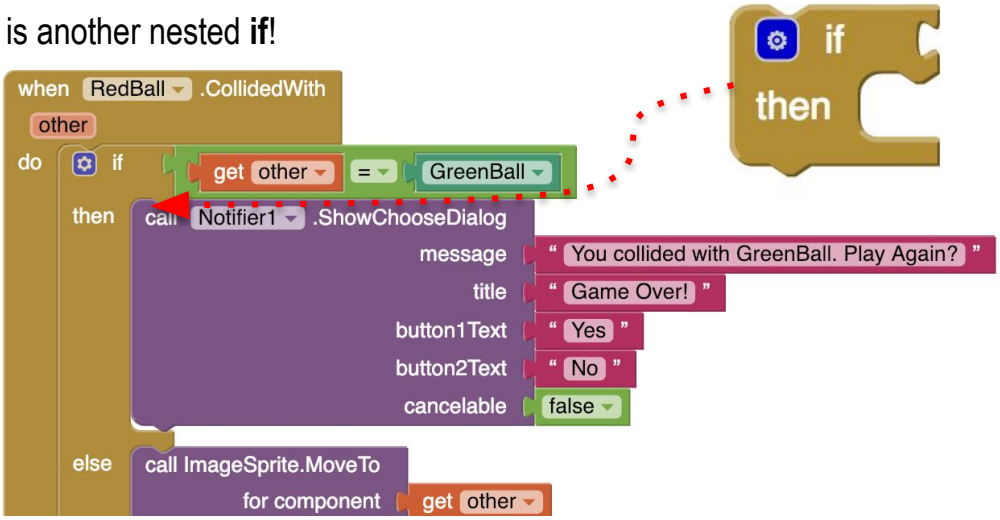


11 Now add to the **RedBall.CollidedWith** block to set the **Score** to be the same as the *Radius* of **RedBall**.

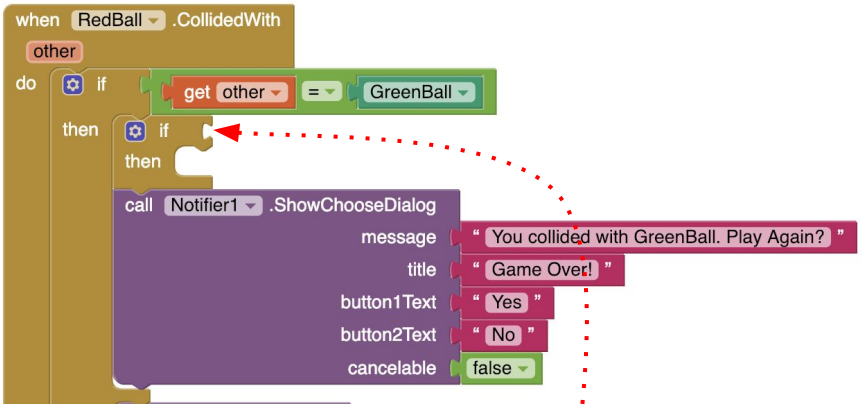


Now for the High Score. When the game ends, test if the current Score is higher than the current High Score. If it is, then the current Score becomes the new High Score.

12 Drag out an **if** block and drag it into the **if get other = GreenBall** block. This is another nested **if**!



SETTING THE HIGH SCORE



13 You want to compare the current Score to the High Score. Drag out an **equals (=)** block from the Math drawer, and change the **=** to **>**, by using the dropdown menu.

Click on the = to get the dropdown and select >.

14 Compare **Score.Text** to **global highScore**.

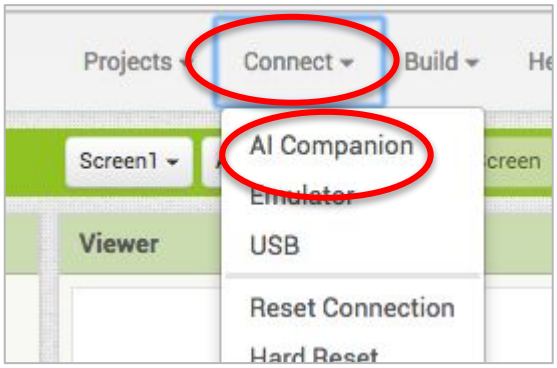
15 If the Score is greater than the High Score, set **highScore** to **Score.Text**. Drag it into the *then* part of the **if** block.

16 And set the **HighScore** label to display the new **highScore**.
Drag the block in below **set global highScore to Score.Text.**

TESTING

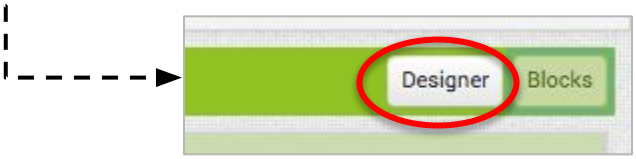
17 Test with the MIT AI2 Companion.

- Try playing the game and see if the high score changes.
- Now close MIT AI2 Companion and run it again. Does the high score display correctly?

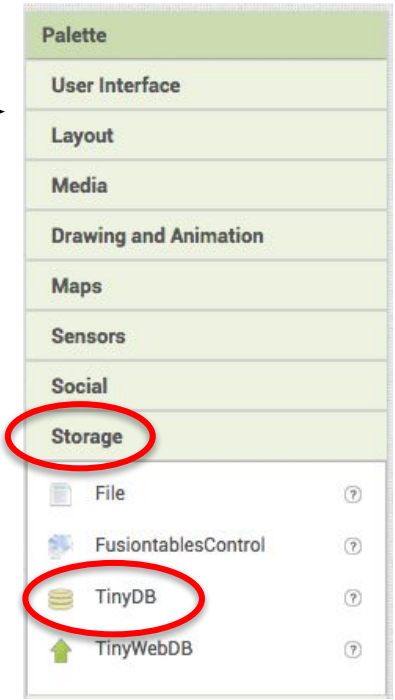


The high score gets set back to 0 because variables are not persistent, which means their values are erased when a program or app closes. You will use a new component, called TinyDB, to save the high score on the device, so it can be saved between different occasions of playing the game.

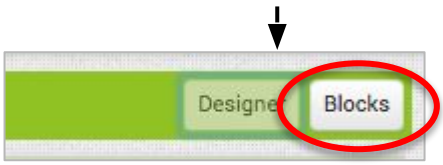
18 Go to the Designer.



19 From the Storage drawer, drag out a **TinyDB** component. It is non-visible, so it will drop to the bottom of the Viewer.



20 Switch back to the Blocks Editor.

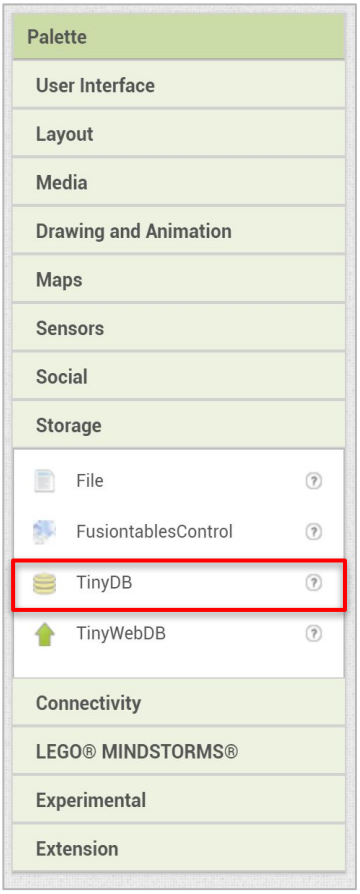


ABOUT TINYDB

Below is an explanation of what TinyDB is and how it works.

TinyDB is a component that stores "persistent" data--so it saves data even after you close the app for the next time you use it. It stands for "Tiny Database".

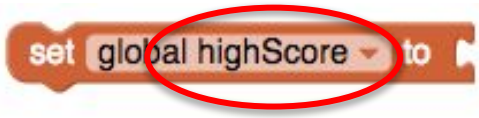
You will use TinyDB to store the high score for the Food Chase app.



TinyDB has two main functions: **StoreValue** and **GetValue**.

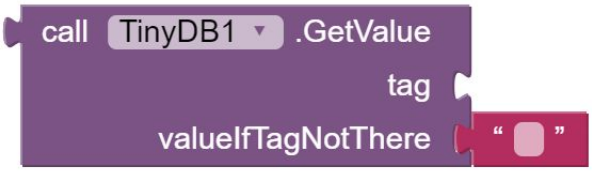
- **StoreValue** stores a value, replacing whatever was in the database before. The name of the value is **tag** and the new value is **valueToStore**.

tag is like a variable name



valueToStore is like a value of a variable

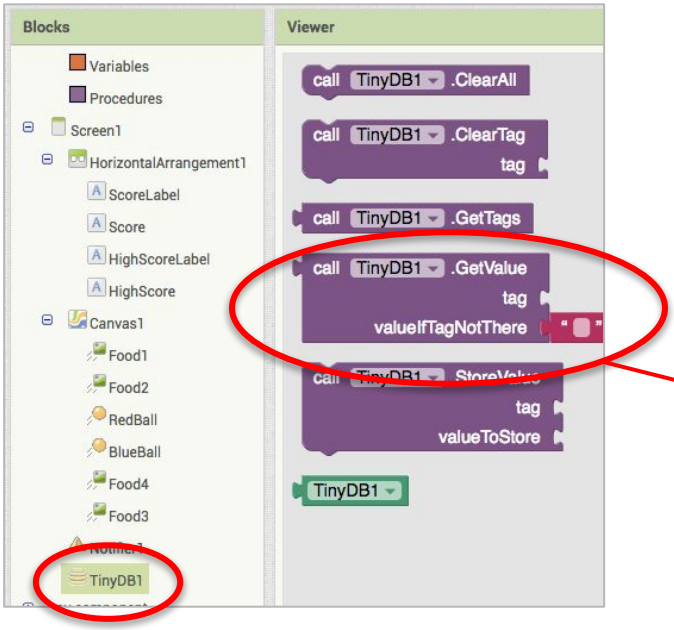
- **GetValue** fetches a value from the database that was stored before, by its **tag**. If there's no value stored, then it returns **valueIfTagNotThere**.



INITIALIZING THE HIGH SCORE

When the app starts, get the high score from **TinyDB**.

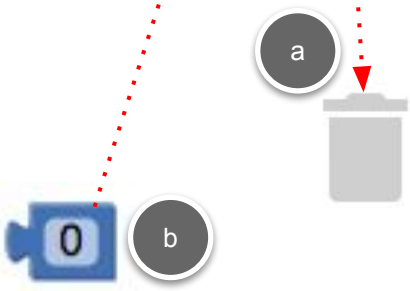
21 Drag out a **TinyDB1.GetValue** block.



Each time you save something to **TinyDB**, you save it with a *tag*. Use "**FoodChaseHighScore**" for your tag.

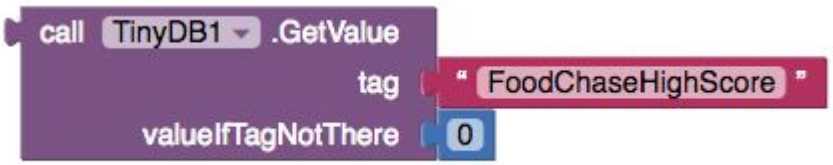
22 Drag out a blank Text block and type in "**FoodChaseHighScore**". Attach it to the tag puzzle piece for **TinyDB1.GetValue**.

23 For the **valueIfTagNotThere**, remove the blank Text block and replace it with a Math **0** block. This lets you use 0 if there is no score yet.

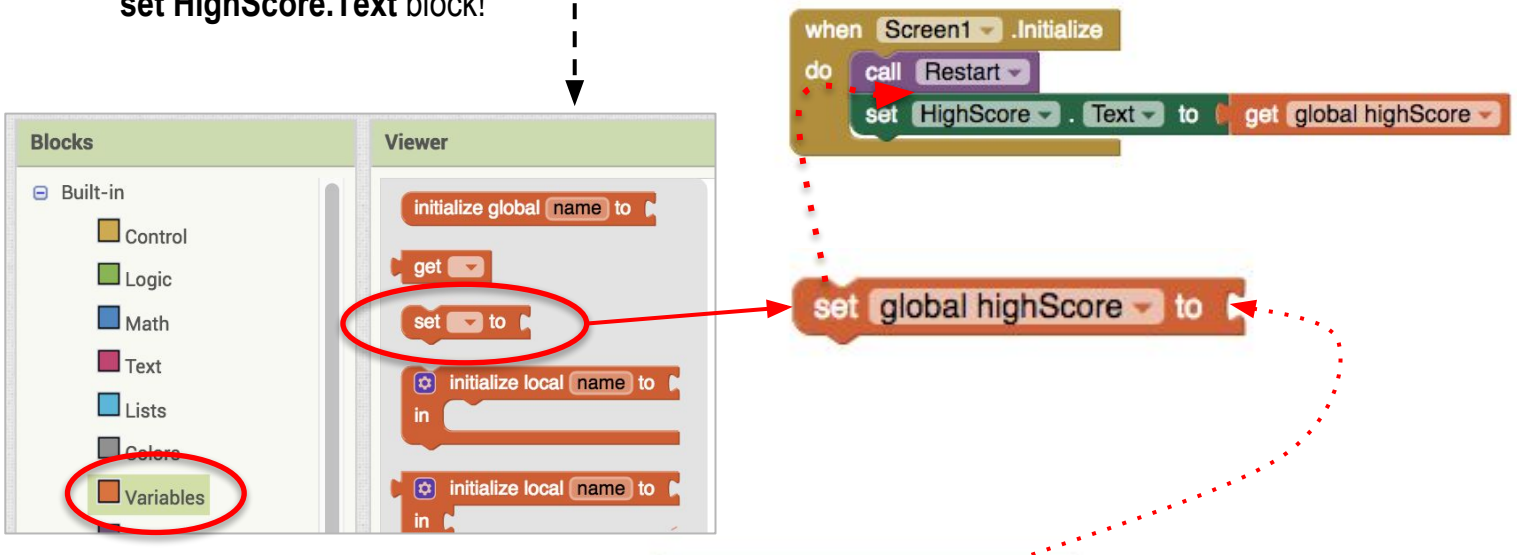


INITIALIZING THE HIGH SCORE

You will add this block to **Screen1.Initialize**. Note that it needs to be snapped into a puzzle piece. What should you do with the high score saved in **TInyDB**?



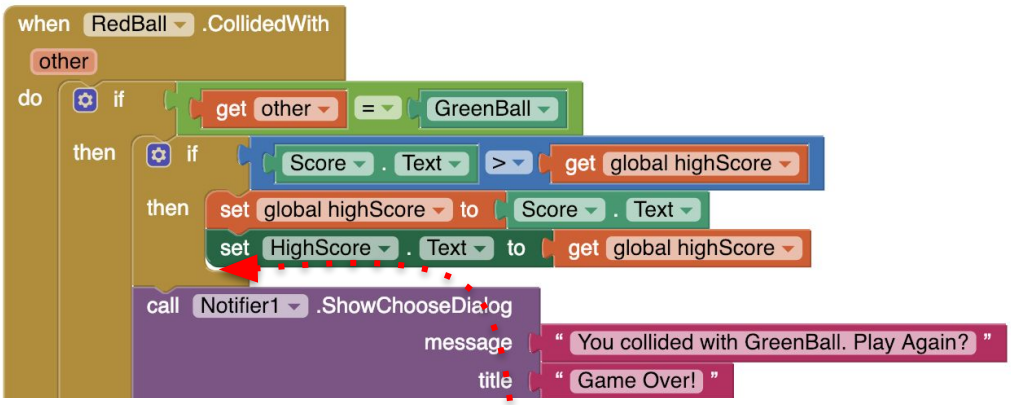
24 Drag out a **set global highScore** block and snap it into the **Screen1.Initialize** block. Make sure to place it before the **set HighScore.Text** block!



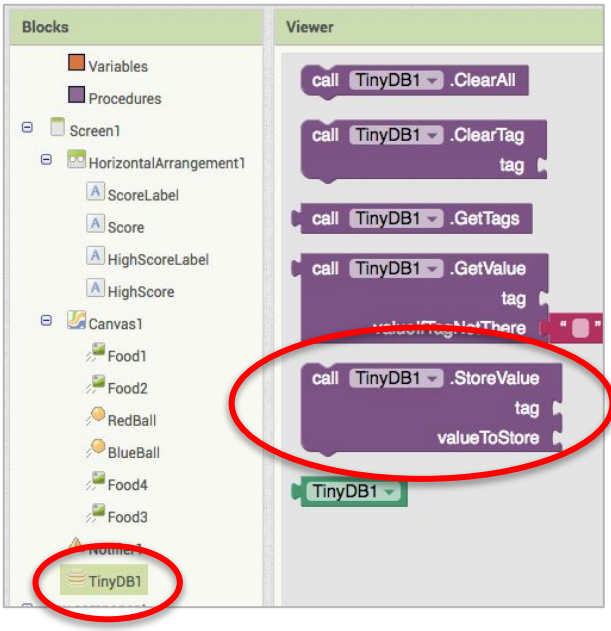
25 Snap the **TinyDB1.GetValue** block to that block.

STORING THE HIGH SCORE

Last, you will store the value of the high score whenever you get a new one.



26 Drag out a **TInyDB1.StoreValue** and snap it into **RedBall.CollidedWith** block where the highScore variable is updated.



27 Use "**FoodChaseHighScore**" for the **tag**.

28 The **valueToStore** is the variable **highScore**.

29 Test again! Now your high score should display correctly, even if you close the app and open it again!

Choose Ways to Extend Your App

Here are a few features you could add if you want to expand your app

Add sounds! One for eating food and another for losing game

Make the GreenBall move faster as time goes by

Make the Food Sprites move too

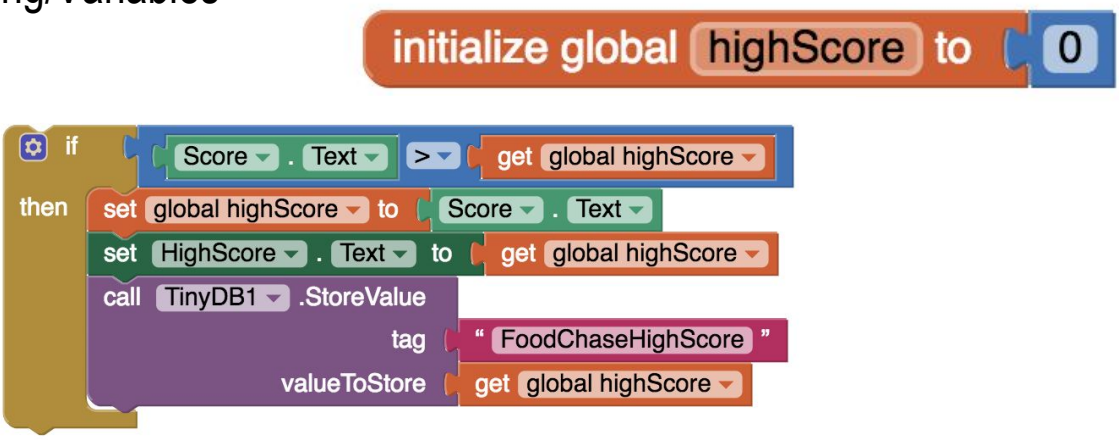
What other ideas do you have?

COMPUTATIONAL THINKING CONCEPTS and PRACTICES

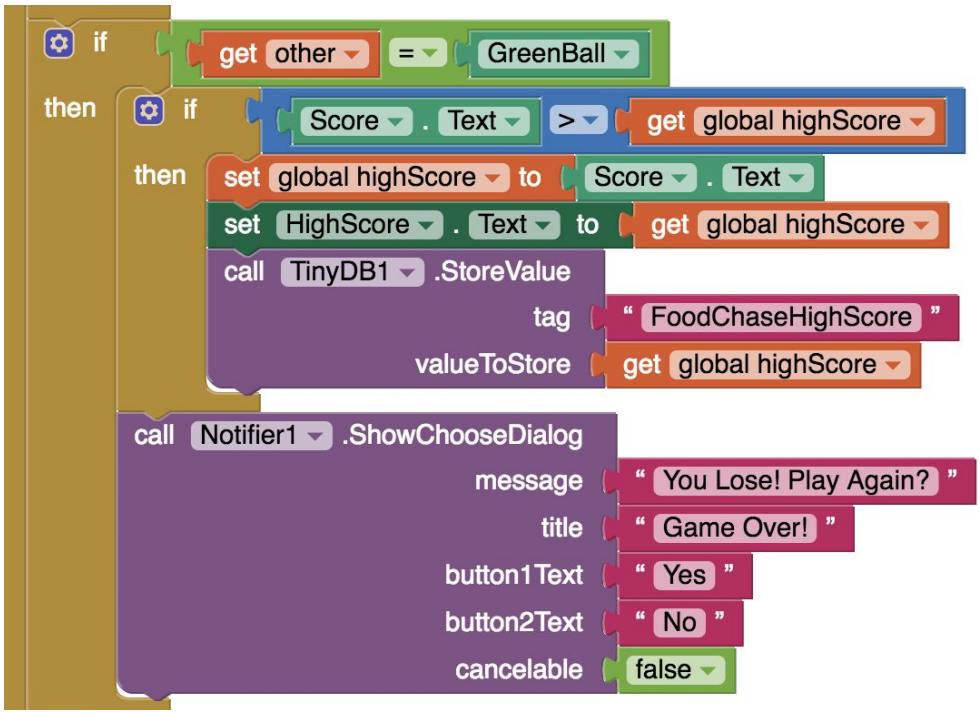
The following are the Computational Thinking Concepts and Practices used in Part 4.

Food Chase Game

1. Naming/Variables



2. Conditionals



COMPUTATIONAL THINKING CONCEPTS and PRACTICES (continued)

The following are the Computational Thinking Concepts and Practices used in Part 4.

Food Chase Game
3. Manipulation of data and elementary data structures 
4. Testing and Debugging ❑ Test again! Now your high score should display correctly, even if you close the app and open it again!