FOOD CHASE GAME: PART 3

You need to fix a few things to make your app work completely and to make it more exciting for users.

- Make **GreenBall** move around the screen so **RedBall** must avoid it.
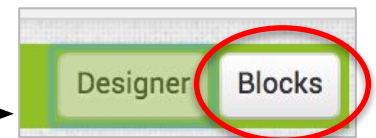- Respond to user selection to the dialog box when **RedBall** and **GreenBall** collide.

In Part 3 you will put some added touches to your app to make it work smoothly for users!
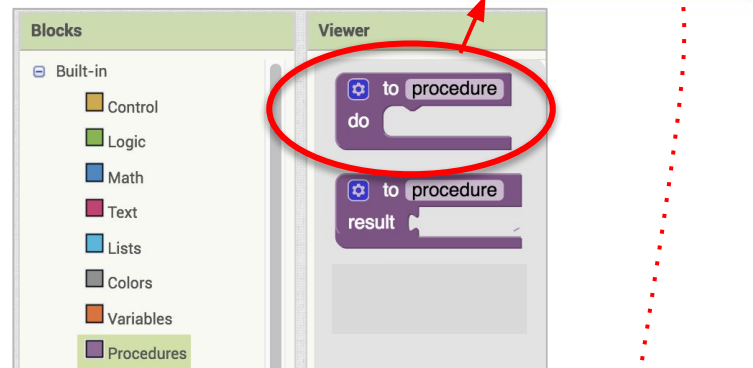
START HERE

1. Open the FoodChase project you made in Part 1 and 2 of this unit, and make sure you are using the Blocks Editor.

Make a procedure that you can use in two places: when the app starts and when the user says Yes to Play Again? in the dialog box.
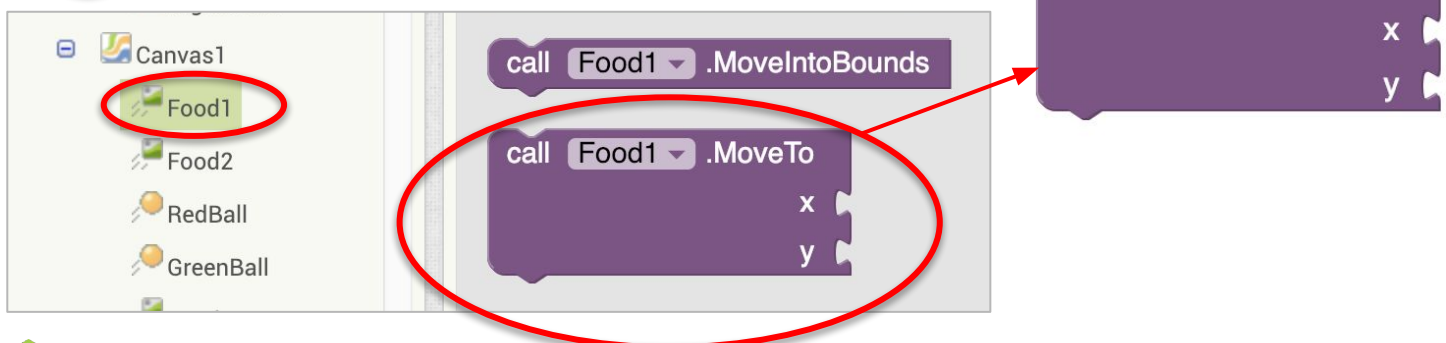
2. Drag out a **do procedure** block from the Procedures drawer and change the name to **Restart**.

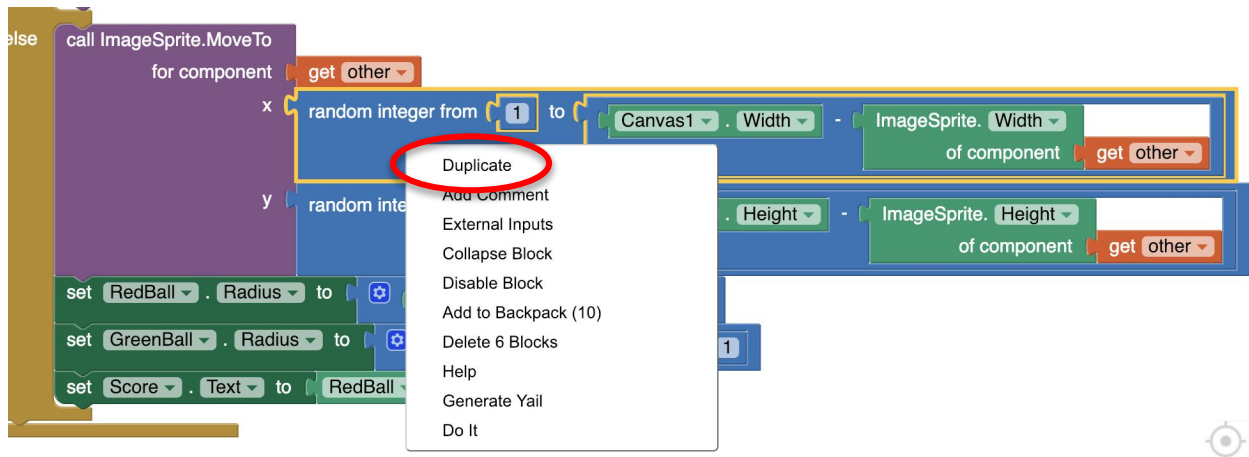You want to randomly place all of the **Food** ImageSprites as well as the **GreenBall**.

3. Drag out a **Food1.MoveTo** block from the **Food1** drawer.
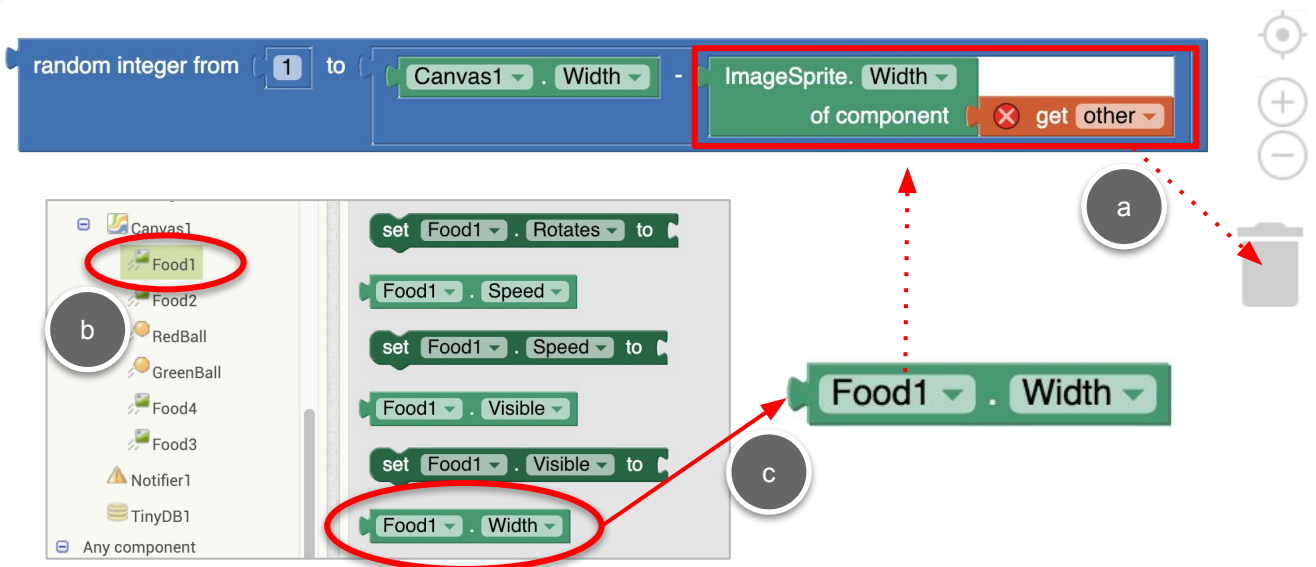
RESTART PROCEDURE

4 The code needed is similar to your random Food placement from **RedBall.CollidedWith.** Duplicate the random integer blocks from **RedBall.CollidedWith** and snap them to the x and y slots here.
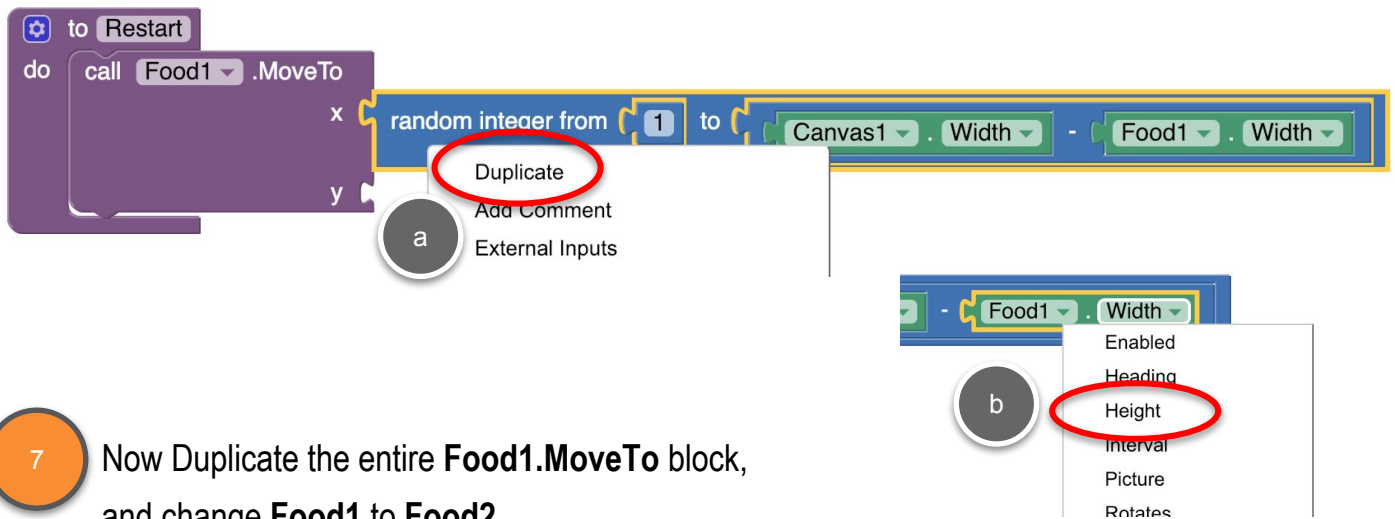

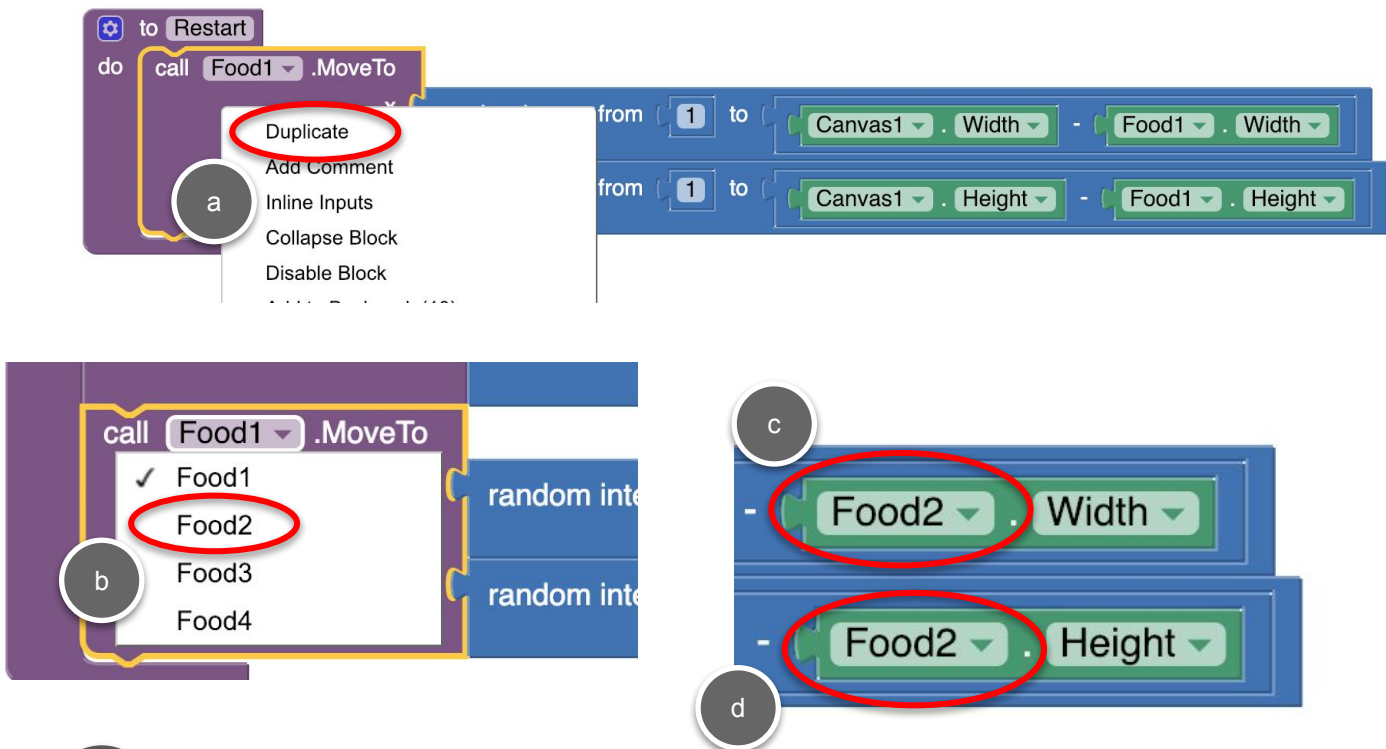**5** However, instead of **ImageSprite.Width**, remove it and use **Food1.Width**.

RESTART PROCEDURE (continued)

6 Now Duplicate the random integer block from the **x** slot and snap the copy into the **y** slot. Remember to change **Food1.Width** to **Food1.Height** though!

7 Now Duplicate the entire **Food1.MoveTo** block, and change **Food1** to **Food2**.

8 Do the same for **Food3** and **Food4.**

MIT APP INVENTOR

RESTART PROCEDURE (continued)

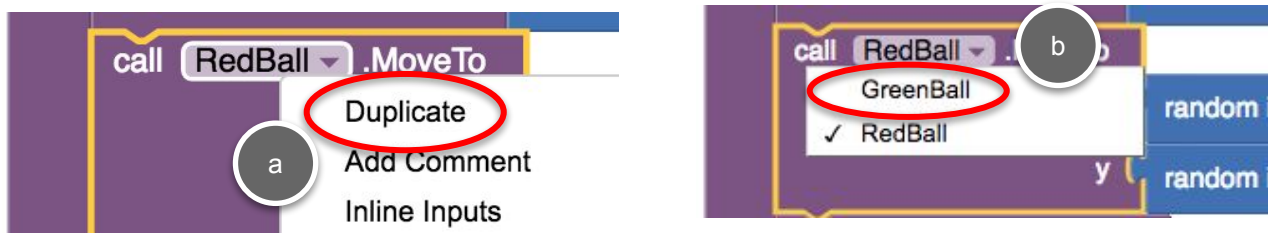**9**   Now place the **RedBall** randomly on the Canvas with a **RedBall.MoveTo** block.

10   Since the **RedBall** is small to start, and will be moving, you can just use a range from 1 to the Canvas' *Width* and *Height* for **random integer**.

11   Duplicate **RedBall.MoveTo**, change to **GreenBall** and snap both **MoveTo** blocks in at the end of **Restart**.

MIT APP INVENTOR

RESTART PROCEDURE (continued)

12 Reset the Radius for **RedBall** to 2.

set RedBall . Radius to

Screen1
Canvas1
Food1
Food2
a
RedBall
BlueBall

set RedBall . PaintColor to

RedBall . Radius

set RedBall . Radius to

b

2 **d**

Blocks

Viewer

Built-in

Control

c Logic

Math

Text

0

13 Duplicate this block so you can do the same for **GreenBall**.

set GreenBall . Radius to 2

14 Also set the speed for **GreenBall** so it will move randomly. Duplicate **set GreenBall.Radius** and change **Radius** to **Speed** and **2** to **5.**

c 5

set GreenBall . Radius to 2

a Duplicate
Add Comment
Collapse Block
Disable Block

set GreenBall . Radius to 2

Enabled
Heading
Interval
b PaintColor
✓ Radius
Speed
Visible

15 Duplicate **set GreenBall.Radius** again and change **Radius** to **Heading**.

set GreenBall . Radius to 2

a Duplicate
Add Comment
Collapse Block
Disable Block

set GreenBall . Radius to 2

Enabled
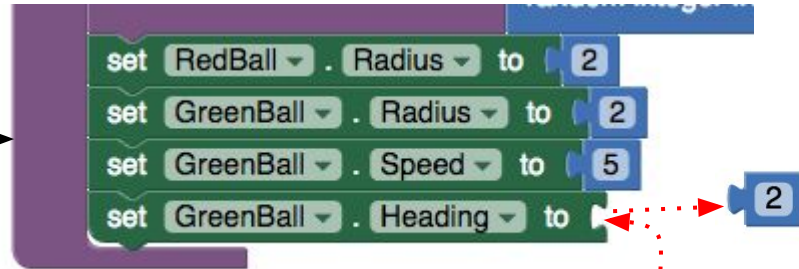**b** Heading
Interval
PaintColor

MIT
APP INVENTOR

RESTART PROCEDURE (continued)

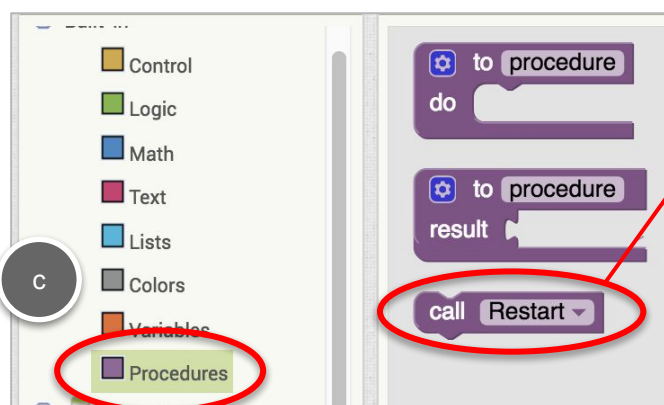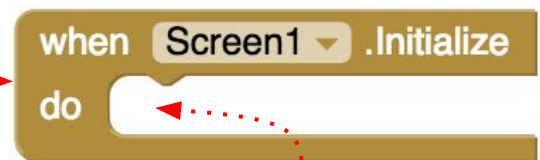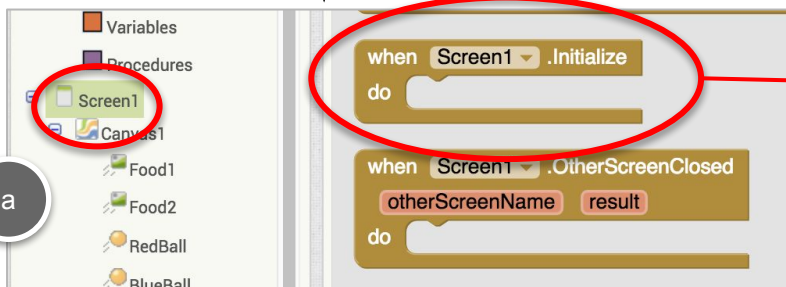**Heading** is the direction so these block make it move in a random direction.

16 Remove the **2** block and replace with a **set random integer** block, and change **100** to **360.**
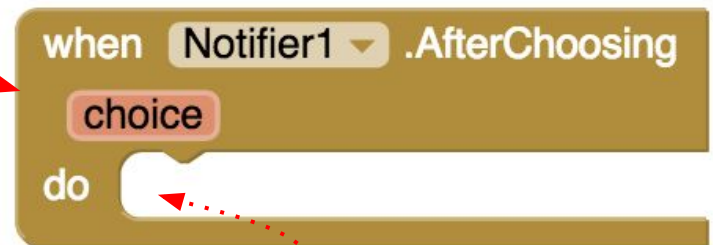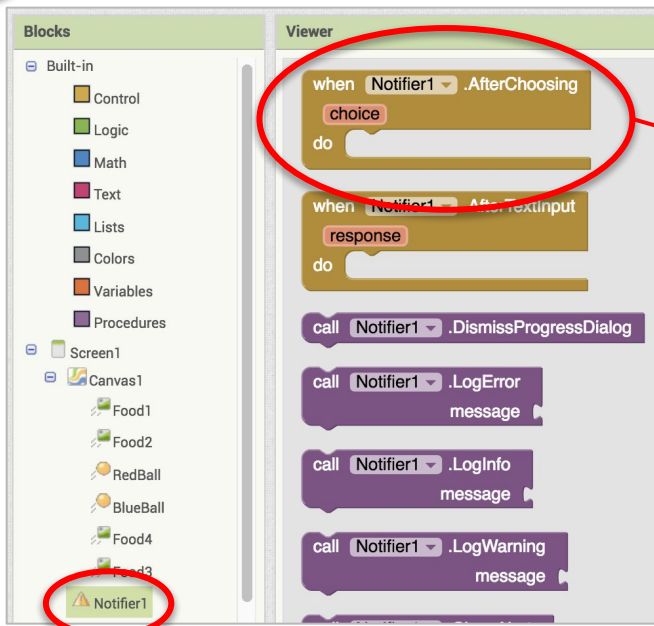
17 All blocks should go at the end of **Restart.**

18 Now that you've made the procedure, you need to add calls to it. You want to call it when the app starts, so drag out a **when Screen1.Initalize** block and add it there.
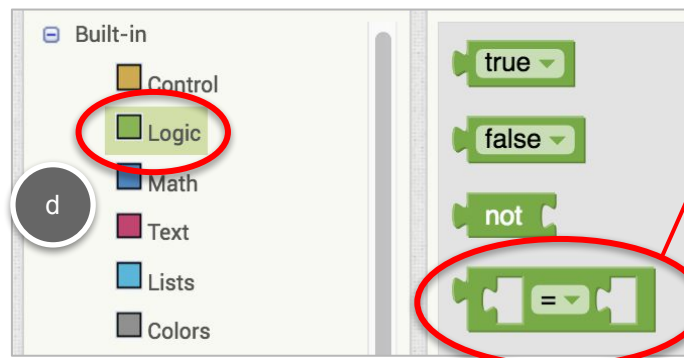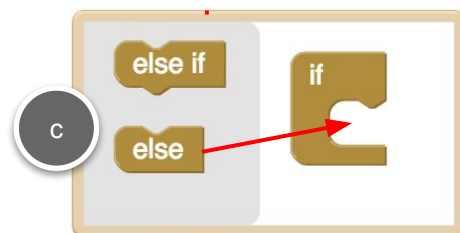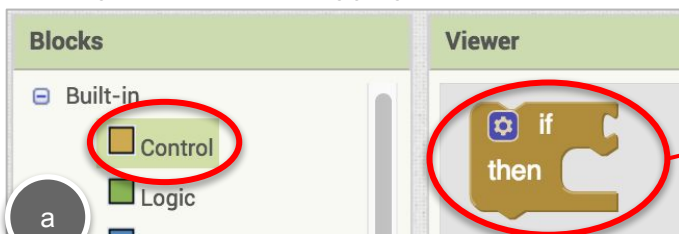
CALL RESTART

The other place to call **Restart** is when the user chooses to Play Again from the Dialog box popup.

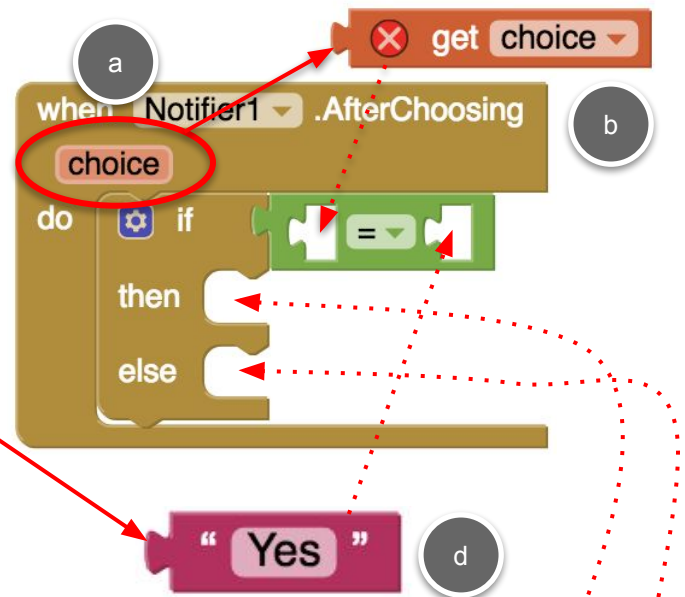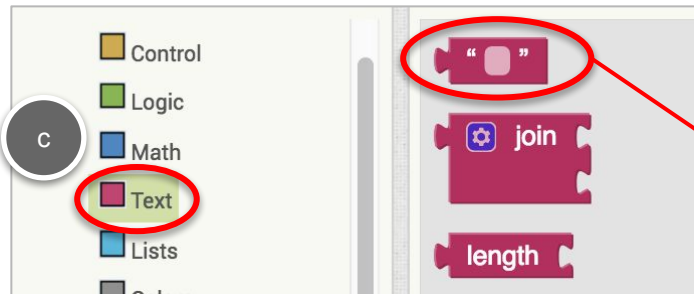**21** Drag out a **when Notifier.AfterChoosing** block.

22 From the Control drawer, drag out an **if** block, and make it an **if-then-else** block by clicking on the blue gear icon and dragging **else** into the block.
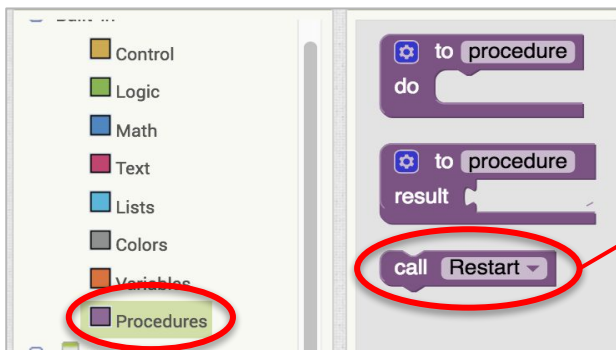
CALL RESTART

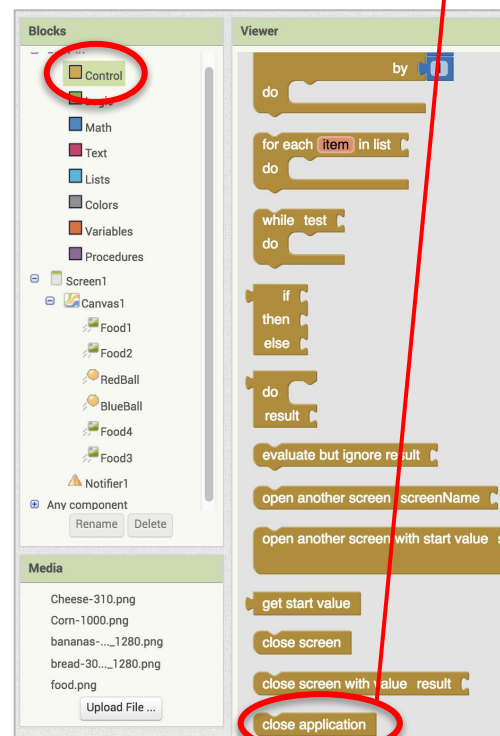**23** Test if the user chose "Yes" to restart the game. - - - - - - - - ▶

24 If the user chose "Yes", call **Restart**.

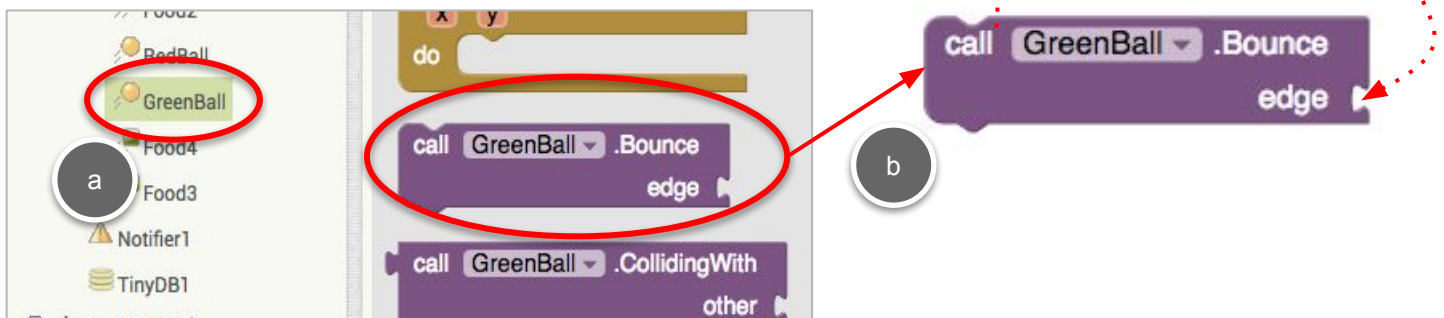**25** Otherwise, close the app. - - - - - - - ▶

BOUNCE GREENBALL

26 Because **GreenBall** is now automatically moving around the screen, you want it to bounce off the edges, not get stuck, so add a **when GreenBall.EdgeReached** block.
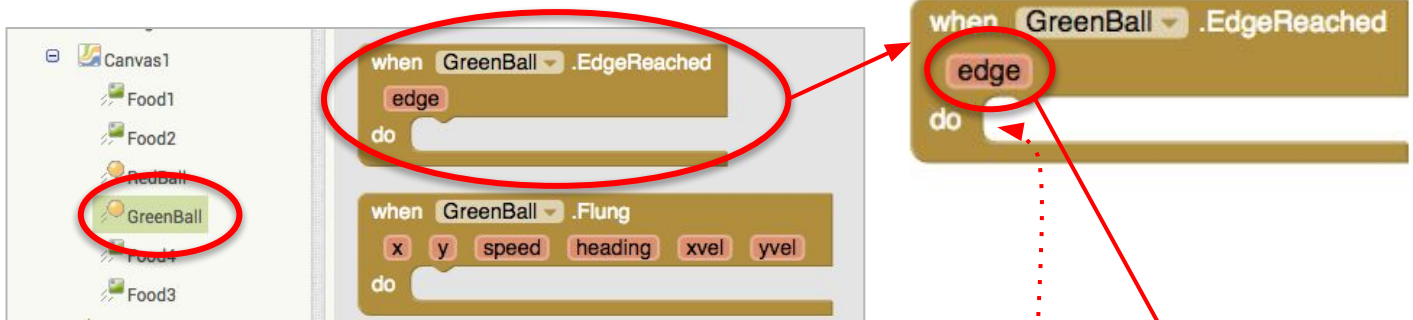
27 Add a **GreenBall.Bounce** block and bounce it off the **edge** that was reached.

28 Now test the app with the MIT AI2 Companion.
- o Does GreenBall move around the screen?
- o Does GreenBall bounce off edges?
- o When the game is over, can you restart by choosing "Yes"?
- o Does choosing "No" close the app? (note you cannot fully test this with the AI2 Companion)
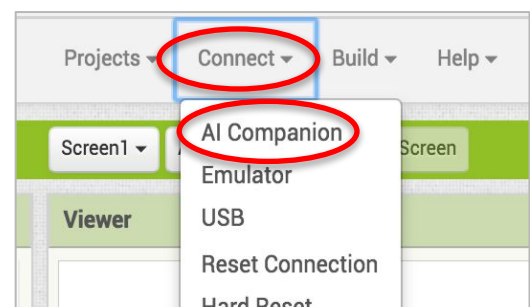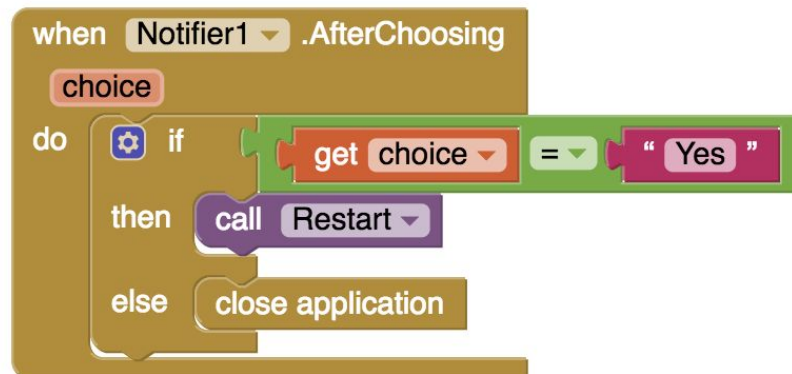
COMPUTATIONAL THINKING CONCEPTS and PRACTICES

The following are the Computational Thinking Concepts and Practices used in Part 3.

Food Chase Game
1. Events
when Notifier1 .AfterChoosing / choice / do — when GreenBall .EdgeReached / edge / do
2. Conditionals
when Notifier1 .AfterChoosing / choice / do / if get choice = "Yes" / then call Restart / else close application
3. Abstraction and Modularization (procedures)
to Restart / do — call Restart