FOOD CHASE GAME: PART 1

In this unit, you will create an animated game about chasing food and growing bigger!
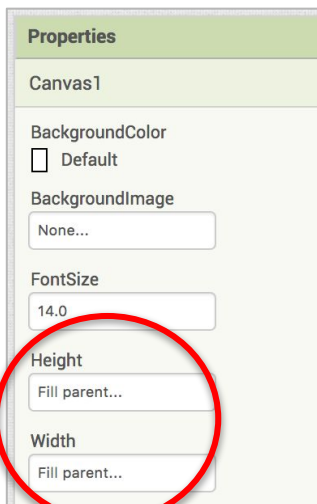
START HERE

1 Login to the MIT App Inventor website (http://ai2.appinventor.mit.edu).
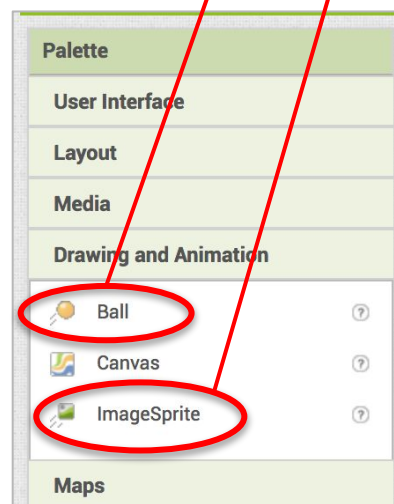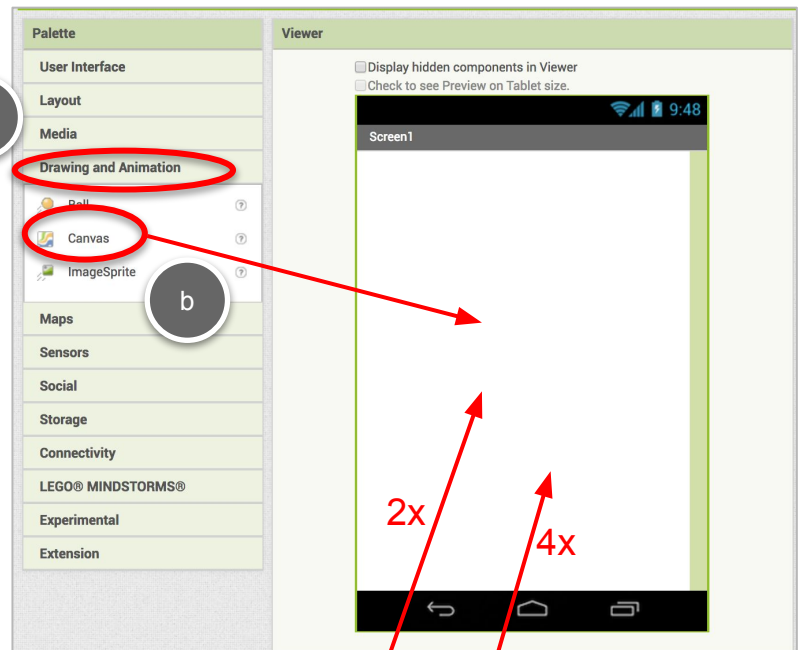
2 Open the FoodChase_template project provided by your teacher.

3 In the Designer window, drag in a Canvas component from the Drawing and Animation drawer.

4 Click on **Canvas1**, and change both its *Width* and *Height* to "**Fill Parent**".

5 From the Drawing and Animation drawer, drag in TWO Balls, and FOUR ImageSprites and drop them on the Canvas.
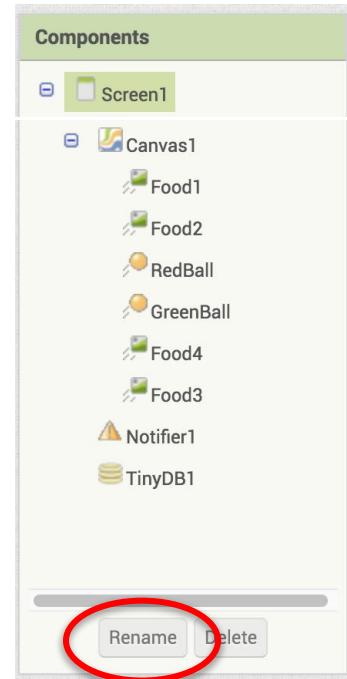
Copyright MIT 2021

GAME SPRITES

For this game, you will have six sprites - 2 **Ball** sprites, and 4 **ImageSprites**. They all work the same way. Ball sprites are automatically round. **ImageSprites** let you change shape and appearance by attaching images.

Components

- Screen1
  - Canvas1
    - Food1
    - Food2
    - RedBall
    - GreenBall
    - Food4
    - Food3
    - Notifier1
    - TinyDB1

Rename   Delete

6 Rename your **Ball** and **ImageSprites** as shown here in the Components panel. Click on each component and click the Rename button to rename it.

7 Look below at the Properties panel for an **ImageSprite** to become familiar with each property.

Properties

Food2

Enabled ☑

Heading 0

Height 30 pixels...

Width 30 pixels...

Interval 100

Picture Cheese-310.png...

Rotates ☑

Speed 10

Visible ☑

X 44

Y 45

Z 1.0

Heading is the direction of the ImageSprite (from 0 to 360 degrees).

Width and **Height** can be set to resize your sprite.

Interval is how often the ImageSprite moves. 100 means every 1/10th of a second. 1000 means every second!

Picture can be set to an image file uploaded to your project.

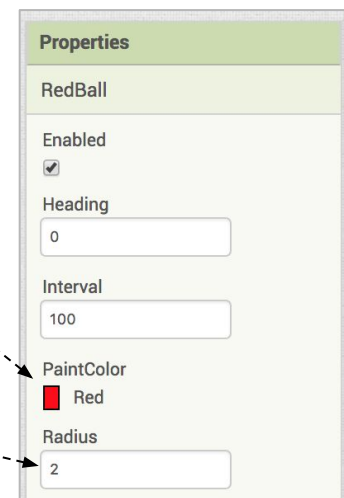**Rotates** here is checked, meaning the ImageSprite rotates according to its heading.

Speed is how fast the ImageSprite moves each Interval. Here it moves 10 pixels.

X and **Y** are the positions of the ImageSprite (before it starts moving).

Z is not used in this app.

8 Note that the Properties for **Ball** components are very similar to **ImageSprites**, except for a few related to color and size.

Properties

RedBall

Enabled ☑

Heading 0

Interval 100

PaintColor ■ Red

Radius 2

PaintColor lets you change the Ball's color.

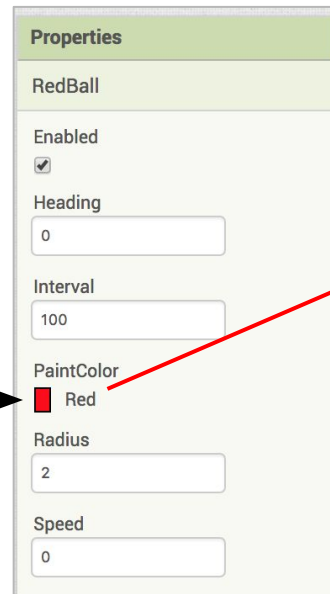**Radius** lets you change the size of the Ball.

MIT APP INVENTOR

SETTING PROPERTIES

9 Click on **RedBall**, and change its properties as shown to the right. To change the *PaintColor*, click on the current color and choose **Red** from the dropdown list.

10 For **GreenBall,** change its properties to be the same as **RedBall**, except of course for its *PaintColor*. Choose **Green** for its *PaintColor.*

11 Click on **Food1** in the Components panel, and change its properties as shown to the right. The other properties can remain the default values.

12 Update the properties for **Food2**, **Food3**, and **Food4** to be the same as **Food1**. Except for *Picture*: choose a different *Picture* for each Food from the dropdown list.

PaintColor
Red

- None
- Black
- Blue
- Cyan
- Default
- Dark Gray
- Gray
- Green
- Light Gray
- Magenta
- Orange
- Pink
- Red
- White
- Yellow
- Custom...

Properties

RedBall
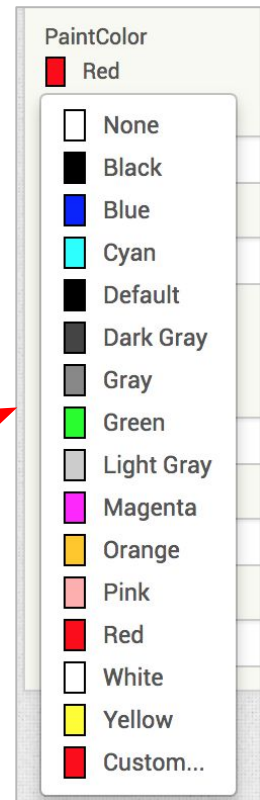
Enabled
☑

Heading
0

Interval
100

PaintColor
Red

Radius
2

Speed
0

Properties

Food1

Enabled
☑

Heading
0

Height
30 pixels...

Width
30 pixels...

Interval
100

Picture
bananas.png...

Rotates
☑

Speed
0.0

Picture
- None
- Cheese-310.png
- Corn-1000.png
- food.png
- bread.png
- bananas.png

Upload File ...

Cancel   OK

Click on None and you will see all the media assets for your project. Choose the banana image file for this ImageSprite.

MIT APP INVENTOR

SCREEN1

13 Set the properties for **Screen1** so the animations appear and work well. Click on **Screen1** in the Components panel, and set its:
- *ScreenOrientation* to **Portrait**
- Uncheck the *Scrollable* property.
- *Sizing* to **Responsive**

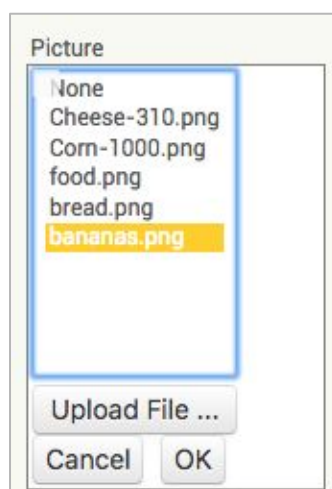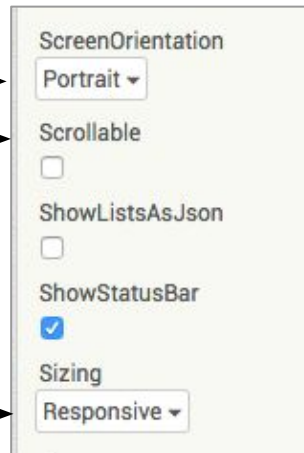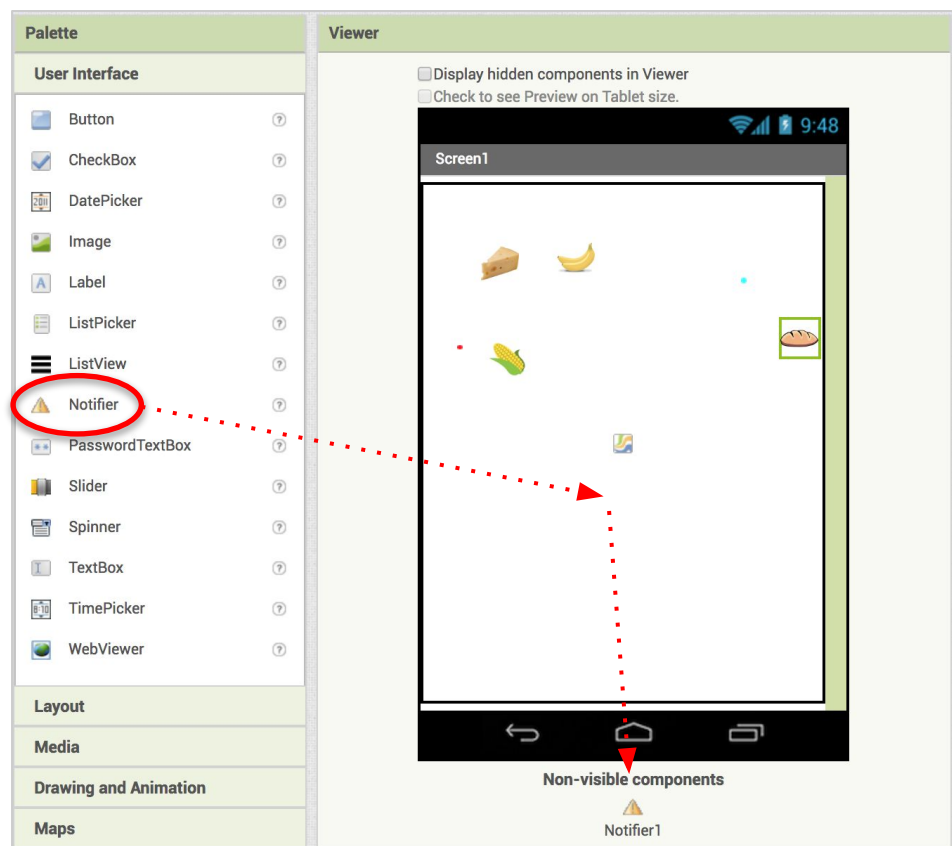**14** The last component to add is a **Notifier** component. Drag it in from the User Interface drawer.

> ScreenOrientation
> Portrait ▾
>
> Scrollable
> ☐
>
> ShowListsAsJson
> ☐
>
> ShowStatusBar
> ☑
>
> Sizing
> Responsive ▾

> *Scrollable* property allow the user to scroll on the screen if checked. No scrolling allowed if unchecked.

> **Responsive** *Sizing* changes the size of components based on the resolution of the device.

> **Notifier** is a non-visible component. It's part of your app, but not seen by the user.

Palette

User Interface

	Button	⑦
✔	CheckBox	⑦
	DatePicker	⑦
	Image	⑦
A	Label	⑦
	ListPicker	⑦
≡	ListView	⑦
⚠	Notifier	⑦
	PasswordTextBox	⑦
	Slider	⑦
	Spinner	⑦
	TextBox	⑦
	TimePicker	⑦
	WebViewer	⑦

Layout

Media

Drawing and Animation

Maps

Viewer

☐ Display hidden components in Viewer
☐ Check to see Preview on Tablet size.

📶 🔋 9:48
Screen1

Non-visible components
⚠
Notifier1

MIT
APP INVENTOR

These orange input parameters are information about the fling action captured by the app.

BLOCKS EDITOR

Designer **Blocks**

15 Switch to the Blocks Editor.

RedBall will be controlled by the user, by a flinging action on the **Canvas**.

when Canvas1 .Flung
x y speed heading xvel yvel flungSprite
do

16 Drag out a **when Canvas1.Flung** event block from the **Canvas1** drawer.

set RedBall . Heading to

set RedBall . Speed to
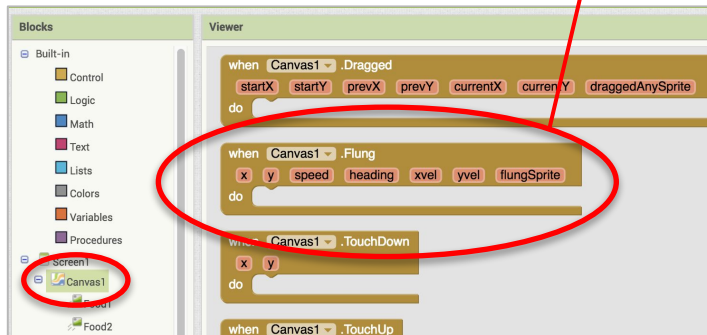
17 From the **RedBall** drawer, drag out **set RedBall.Heading** and **set RedBall.Speed** blocks and snap them into **when Canvas1.Flung.**

You could also use **RedBall.Flung** here. Using **Canvas1.Flung** means the user doesn't have to exactly touch **RedBall** on the flinging action to get it to move.

MIT APP INVENTOR

SET REDBALL PROPERTIES

You want to set the direction (heading) and speed according to the fling action of the user.
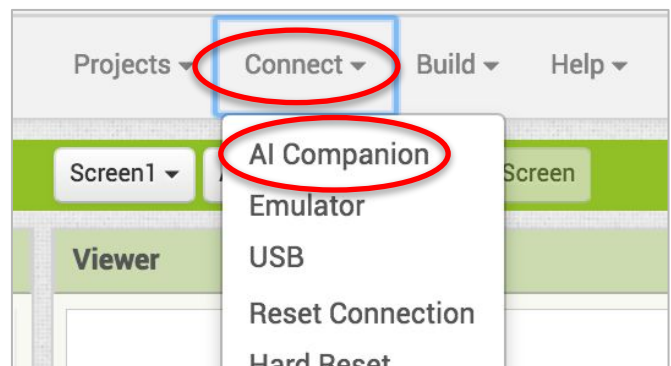
18 Hover over **heading**, and snap **get heading** to the **set RedBall.Heading** block.

19 Hover over **speed**, and snap **get heading** to the **set RedBall.Heading** block.
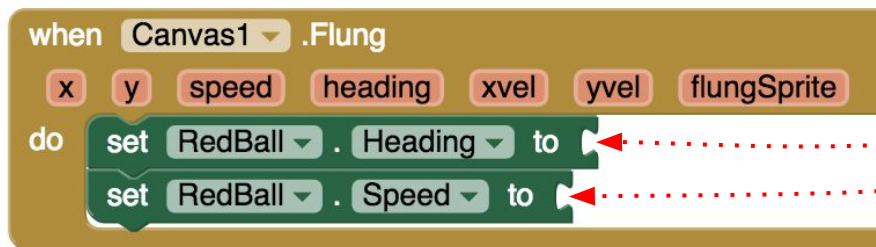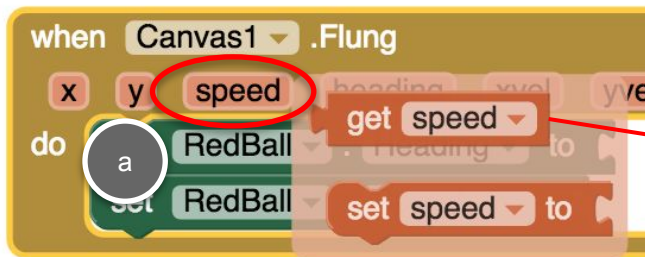
20 Try that out with MIT AI Companion!
Start MIT AI companion on your device.
Try flinging the red ball. It should respond to your fling actions.

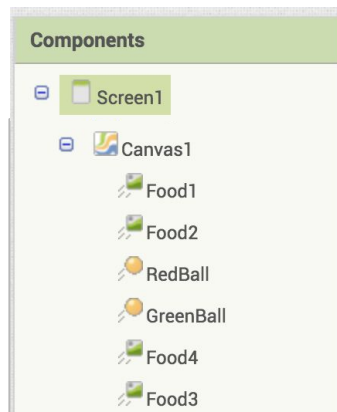Next, you'll add interactions with the other sprites.

6

Copyright MIT 2021

COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts learned in Part 1.

Food Chase Game
1. Naming:
2. Events: