

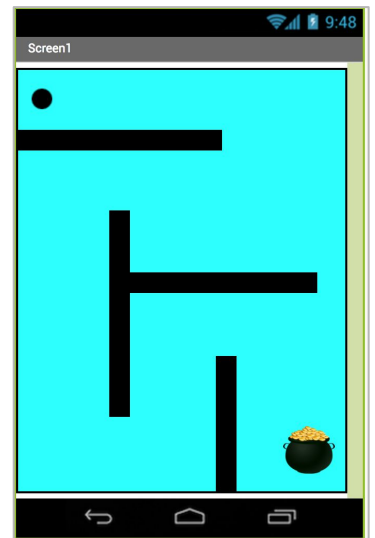
FIND THE GOLD: PART 1

START HERE

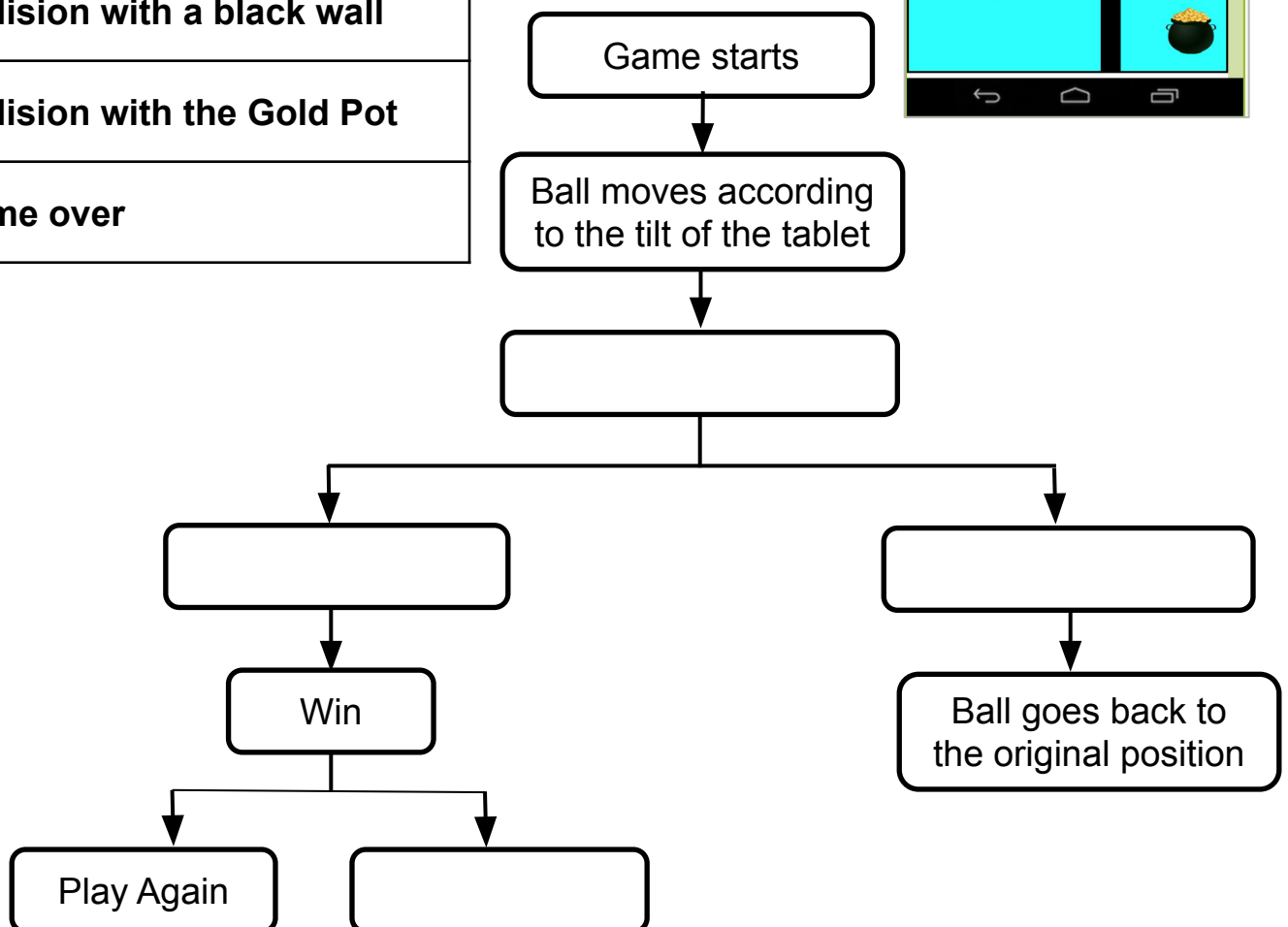
You will make a new game app that moves a ball through a maze when you tilt your smartphone or tablet.

1

Look at and discuss the following diagram with your partner. Try to complete the missing steps for this app by selecting A, B, C or D below. Some have been filled in for you.



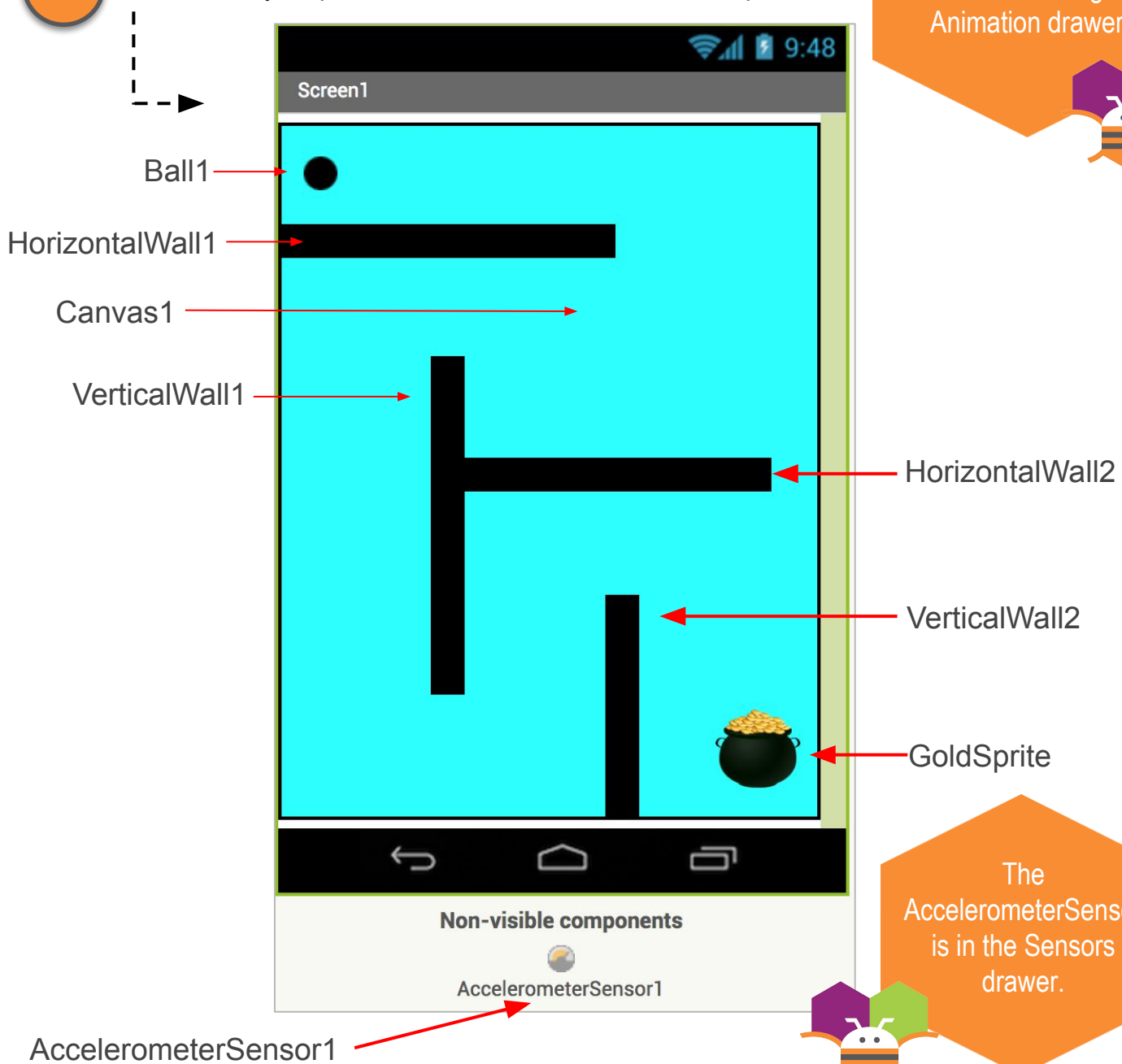
A. Ball collision detected
B. Collision with a black wall
C. Collision with the Gold Pot
D. Game over



THE DESIGNER

- 2 Open the **FindTheGold** template project, provided by your teacher, in MIT App Inventor.
This is what it should look like in the Designer.

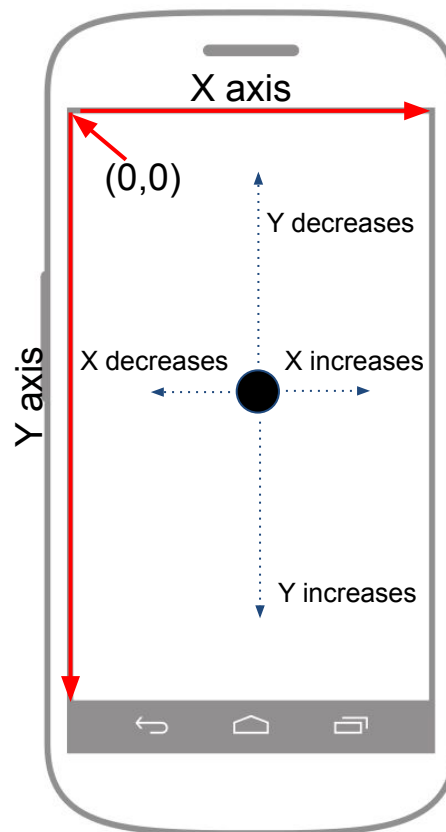
- 3 Discuss with your partner the function of each of the components.



IMAGESPRITE AND BALL COMPONENTS

In the game app, the walls are **ImageSprites**. **ImageSprites** are sprites that are represented by images in the app. While **ImageSprites** can be animated, the maze walls will remain stationary for this game. **Ball** sprites are automatically round. You can change the color and size of a **Ball** component.

You can set the position of **ImageSprites** and **Balls** by their **X,Y** coordinates, just like in the Cartesian coordinate system. One difference is that the origin (0,0) is at the top left corner of the screen. X increases as it moves to the right of the screen. Y increases as it moves down the screen. - - - - - ➔



ImageSprite Properties

Height and Width can be set to resize your sprite.

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

X and Y are the positions of the ImageSprite.

Properties	
HorizontalWall1	
Enabled	<input checked="" type="checkbox"/>
Heading	0
Height	20 pixels...
Width	200 pixels...
Interval	100
Picture	horizontalwall.jpg...
Rotates	<input checked="" type="checkbox"/>
Speed	0.0
Visible	<input checked="" type="checkbox"/>
X	-5
Y	56
Z	1.0

Ball Properties

PaintColor lets you change the Ball's color.

Radius determines the size of Ball.

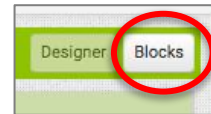
Properties	
RedBall	
Enabled	<input checked="" type="checkbox"/>
Heading	0
Interval	100
PaintColor	Red
Radius	2

SET UP THE MAZE

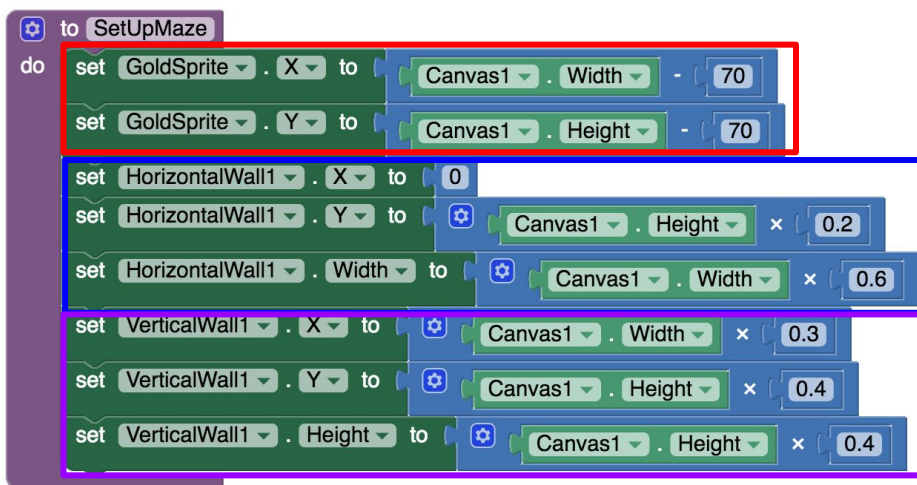
Different mobile devices have different **resolutions**, which means different numbers of pixels on the screen. Phones differ in size, and tablet sizes differ too. So, the size and position of the maze walls must be adjusted to take into account the resolution, or size of the mobile device screen.

To take care of that, the FindTheGold template includes a procedure, called **SetUpMaze**, that places the Wall **ImageSprites** and the **GoldSprite** on the **Canvas** using percentages of the Canvas size. That way, if the Canvas (or Screen) is larger, the size and placement scales up too.

- 4 Switch to the Blocks Editor. ----->



- 5 Find the **SetUpMaze** procedure and look at the blocks. Can you see how the percentages of the **Canvas Height** and **Width** are used to place the **ImageSprites**?

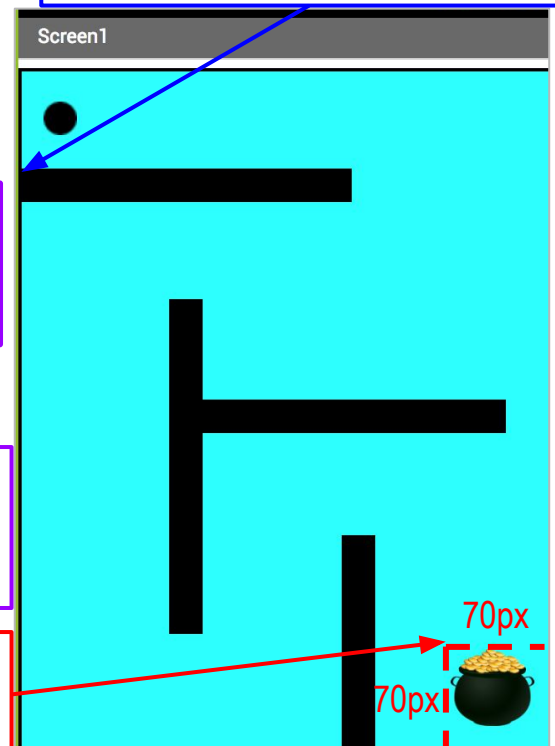


X is 0
Y is 20% of the Canvas Height
Width is 60% of the Canvas Width

The top left corner of each **ImageSprite** is where its X,Y coordinate is located.

X is 30% of the Canvas Width
Y is 40% of the Canvas Height
Height is 40% of Canvas Height

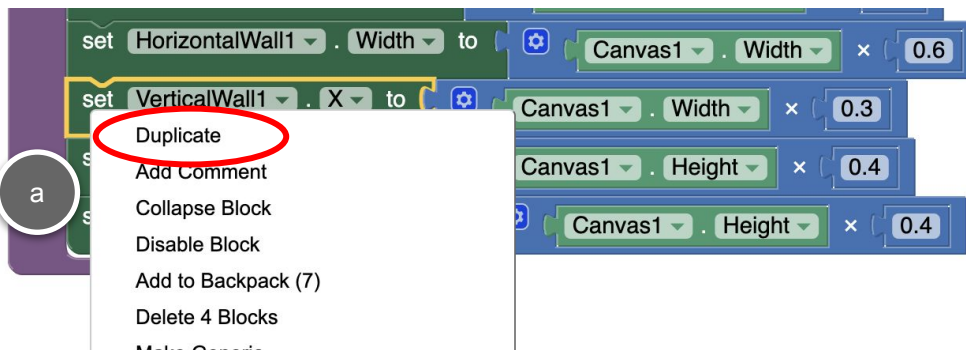
X,Y is 70 pixels from the left bottom corner of the Canvas



SET UP THE MAZE

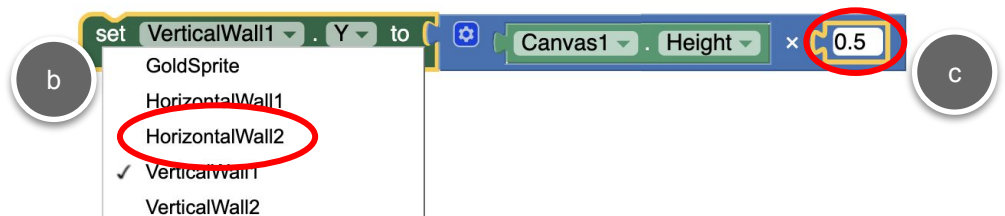
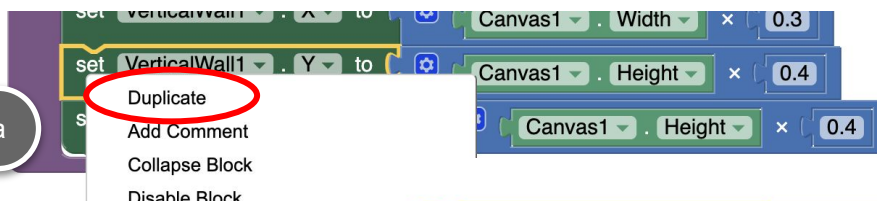
But wait! **HorizontalWall2** and **VerticalWall2** haven't been placed yet. Using **HorizontalWall1** and **VerticalWall1** as examples, add code blocks for **HorizontalWall2** and **VerticalWall2** so that the placement and size work well for your maze.

- 6 An easy way to do this is to Duplicate a corresponding block, and change the parts. For example, Duplicate **set VerticalWall1.X**, and then using the dropdown, change **VerticalWall1** to **HorizontalWall2**.



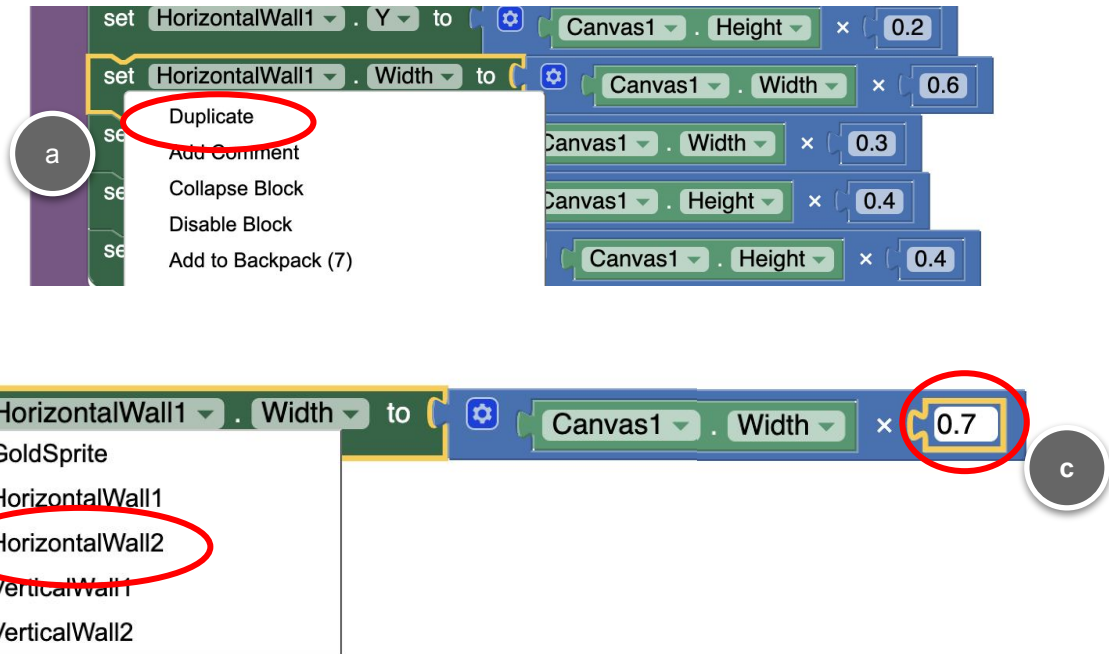
Leave the rest of the block the same, so **VerticalWall1** and **HorizontalWall2** will have the same X.

- 7 Now, duplicate **set VerticalWall1.Y** and change **VerticalWall1** to **HorizontalWall2**, and then change 0.4 to 0.5, to move it down a bit.

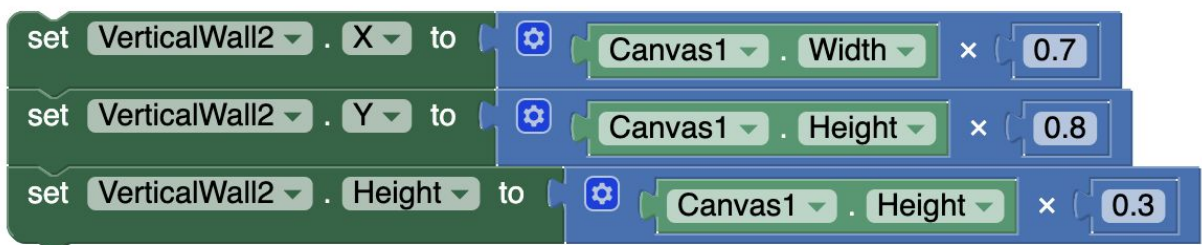


SET UP THE MAZE (continued)

- 8 Now, duplicate **set HorizontalWall1.Width**, change **HorizontalWall1** to **HorizontalWall2**, and change 0.6 to 0.7



- 9 Do the same to set the **X,Y**, and **Height** for **VerticalWall2**. Duplicate existing blocks, and change values using the dropdown menus and by typing. The result should look like the blocks below.



- 10 Snap all the duplicate blocks in under the existing ones in the SetUpMaze procedure.

CALL SETUPMAZE

- 11 The **SetUpMaze** procedure must be called when the app starts, which is in the **Screen1.Initialize** event. Drag out that event block, and snap in a **call SetUpMaze** block.

The screenshot shows the MIT App Inventor interface. On the left, the 'Blocks' palette is open, and the 'Procedures' category is selected (labeled 'c'). In the center, the 'Viewer' pane shows two event blocks: 'when Screen1.Initialize' (labeled 'b') and 'when Screen1.OtherScreenClosed'. A red circle highlights the 'call SetUpMaze' block in the 'Procedures' category (labeled 'd'). A red arrow points from this block to the 'do' slot of the 'when Screen1.Initialize' block. On the right, a separate view shows the 'when Screen1.Initialize' block with the 'call SetUpMaze' block snapped into its 'do' slot (labeled 'a').

- 12 Connect to MIT AI2 Companion on your mobile device, and check how your maze looks! If it doesn't look quite right, try changing the numbers in your code blocks. To check your changes, you can use the **Do It** command. Right click on **call SetUpMaze** and select **Do It**.

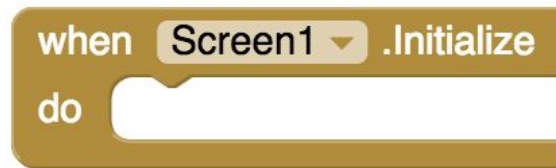
The screenshot shows the MIT App Inventor interface. At the top, the 'Connect' button is highlighted (labeled 'a'). Below it, the 'AI Companion' option is selected from the dropdown menu (labeled 'a'). On the right, a context menu is open, showing the 'Do It' option selected (labeled 'b'). The context menu also includes options like 'Duplicate', 'Add Comment', 'Collapse Block', 'Disable Block', 'Add to Backpack (0)', 'Delete Block', 'Help', 'Generate Yail', and 'Highlight Procedure'.

COMPUTATIONAL THINKING CONCEPTS

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

Find the Gold

1. Events:



2. Naming: ImageSprites

