

1 st grade (E.S.O.) UNIT 6: PROGRAMMING (IT ACTIVITIES 1/2)	
1 NAME AND SURNAMES:	Group
2 NAME AND SURNAMES:	

Day/Date	Signatures		Day/Date	Signatures		Day/Date	Signatures	
1/	Name1:	Name2:	2/	Name1:	Name2:	3/	Name1:	Name2:
4/	Name1:	Name2:	5/	Name1:	Name2:	6/	Name1:	Name2:
7/	Name1:	Name2:	8/	Name1:	Name2:	9/	Name1:	Name2:

1 **On-line activities.** Go to the blog of our subject, click on *TPR1 General Website...* and complete the 3 first **ICT activities** of Unit 6.

Programs: algorithms 1  2  3 (EXTRA) 

2 **Introduction to Programming.** Go to our Virtual Classroom, find the **IT_Activities_1/2** for Unit 6 and complete them.

ACTIVITY	SIGNATURE	ACTIVITY	SIGNATURE	ACTIVITY	SIGNATURE
01_1		03		06	
01_2		04		06 EXTRA 3	
02_1		05 <small>HLEVEL</small>		07	
02_2		05 EXTRA 2		08 <small>HLEVEL</small>	

Programming: Understanding Scratch

3 **Moving around the stage 1.** For this program you can choose 2 or 3 different **sprites** or a sprite with more than one looks (**costume**) so that you create a kind of animation (something that impresses).

For instance you can choose a  ,  ,  or a  In this

exercise there are 3 **scripts**, but you may create one single program). When finished call the **teacher**.

- **01:** To give the impression of movement, we need to use repetition to constantly switch between **costumes**. We do this using a **forever loop**. Let's make the horse move forward (→)
- **02:** Next, we need to move the sprite when certain keys (from **Sensing**) are pressed (keyboard input). So, if we press the **right arrow**, we make the horse **turn ↶ 15 degrees**. Repeat this process pressing **left arrow** to make the horse **turn ↷ 15 degrees**.
- **03:** Repeat Script01 to make the parrot fly forward and when **if on edge, bounce**

Script01

```

when green flag clicked
  go to x: 0 y: 0
  forever loop
    next costume
    wait 0.5 secs
    move 10 steps
  
```

Script02


```


when green flag clicked
  forever loop
    if key right arrow pressed? then
      turn ↶ 15 degrees
    if key left arrow pressed? then
      turn ↷ 15 degrees
  
```

Script03






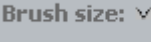


4 Making the sprite move across the stage.

- Select a new sprite (the ghost) from the category Fantasy 
- Create the programs below and complete the sentences:

Motion1	Motion2	Motion3 (4 scripts)
<p>First, shrink the sprite (get smaller) </p> <pre> when green flag clicked forever loop go to mouse-pointer </pre>	<pre> when green flag clicked go to x: 0 y: 0 glide 1 secs to x: 220 y: 160 glide 1 secs to x: 220 y: -160 glide 1 secs to x: -220 y: -160 glide 1 secs to x: -220 y: 160 glide 1 secs to x: 220 y: 160 glide 1 secs to x: 0 y: 0 </pre>	<pre> when green flag clicked forever loop if key right arrow pressed? then point in direction 90 move 10 steps </pre>
<p>Motion3: create the whole program to move the sprite in all directions (right: 90°, down: 180°, left: -90° and up: 0°).</p>	<p style="text-align: center;">Motion4</p> <pre> when green flag clicked go to x: 0 y: 0 clear pen down set pen size to 5 set pen color to red forever loop set pen color to red point towards mouse-pointer move 10 steps if key space pressed? then clear </pre>	

5 Drawing a maze (labyrinth).



• Firstly, we need to draw our maze. Click on the stage icon  /  and then select **Backdrops** / **Backgrounds** (scenery).  Choose the straight line  and select a colour (red) and a size  / . Draw your maze and ensure it has a coloured border all the way around the edge (border).

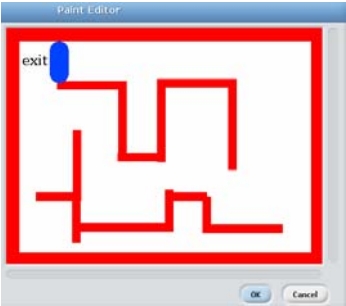
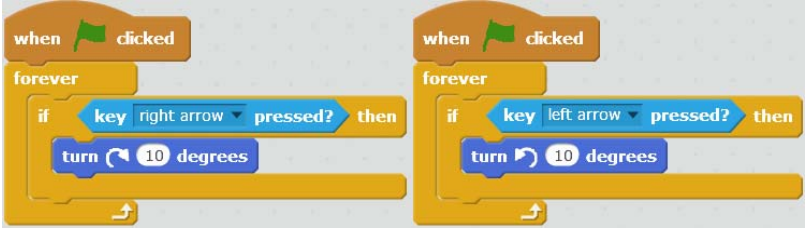
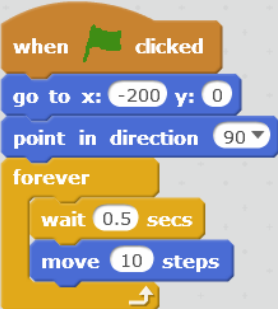

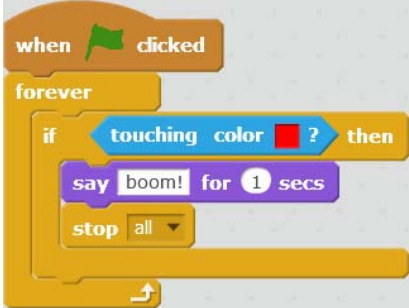
• The sprite (the cat) is probably too large (big) to move around the maze, so you need to



shrink it:  → 


• **First step:** we make the sprite **move forwards** starting at **x: -200 y: 0** (script below)

• **Second step:** we make the sprite **turn up and down** using the left and right arrows


• **Third step:** the sprite cannot go through the maze's walls, so we have to check if (whether) the sprite is touching the wall. When the **sprite touches the wall**, we will say "Boom!" for 1 second and the game will end (finish). To detect the wall follow the instructions below **if touching color  ? then**. Make sure you have activated , play the game and call the **teacher**.

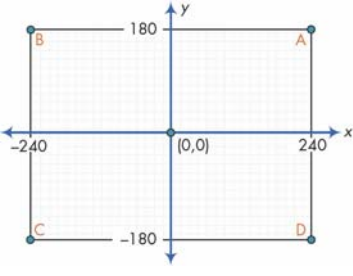
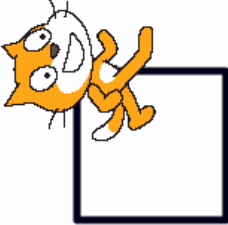
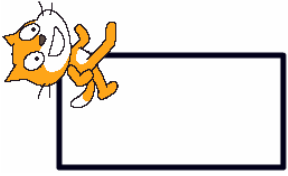
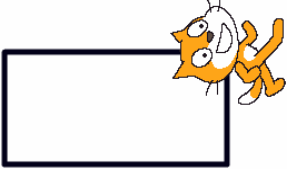
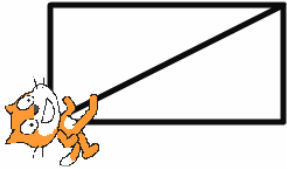

<p style="text-align: center;">Maze</p> 	<p style="text-align: center;">Second step</p> 
<p style="text-align: center;">First step</p> 	<p style="text-align: center;">Third step</p> <p style="text-align: center;">if touching color  ? then</p> <p style="text-align: center;">Click on the colour within (inside) the touching colour (red), followed by clicking on your maze outline on the stage (this will alter the touching colour to be the same as your maze)</p> 

- **Extra point:** make the cat get to the exit of the maze by hitting something (in a different colour)  and saying Hooray! I am the winner!  Call the **teacher_2** (CTT).

- **Extra at home.** Try a different version of the maze and a bit more challenging 

6 **Drawing.** Write the following program and then modify it to draw different figures.

- **original:** indicate the coordinates of each vertex (x, y)
- **mod1** and **mod2:** you only have to modify the x-coordinate, starting at (0,0)
- **mod3:** taking into account where the cat is (mod2) you can make the cat draw the diagonal (go to the opposite vertex). **Call the teacher.**
- **Extra point.** Make the sprite draw the official flag of Galicia (mod4) and  the cat.
- **Extra at home.** Draw your own flag

<pre> when green flag clicked clear go to x: 0 y: 0 set pen size to 5 set pen color to black pen down point in direction 90 move 100 steps point in direction 180 move 100 steps point in direction -90 move 100 steps point in direction 0 move 100 steps pen up </pre>		<p>original</p> 	<p>mod1</p> 
	<p>mod2</p> 	<p>mod3</p> 	<p>mod4 (extra)</p> 

7 **Drawing polygons** (triangles, squares, etc.) and **circles**. When finished, call the **teacher**.

- **O1** Create the **script below**, run the program and **draw the figure** the cat draws.

<pre> when green flag clicked clear set pen size to 6 set pen color to red pen down </pre>	<p>O1 (polygon1)</p>	<p>O2 (polygon2)</p>	<p>O3 (polygon3)</p>
--	-----------------------------	-----------------------------	-----------------------------

	04 (polygon2)	05 (polygon2)	06 circle
--	---------------	---------------	-----------

- 02 Change the size (10) and the colour (blue) of the pen and the side (300 steps)
- 03 Draw a square with 200 step sides
- 04,05 Draw a pentagon/hexagon with 110 step sides
- 05 Draw a circle of 1 step radius

```

repeat 1
  move 10 steps
  turn 90 degrees
  wait 1 secs

```

```

repeat 5
  move 110 steps
  turn 72 degrees
  wait 1 secs

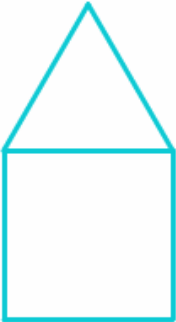
```

```

repeat 360
  move 1 steps
  turn 1 degrees

```

- Extra point. Draw a circle twice as big as the last one and draw it at the other side
- 8 Draw the house below using any colour (the house is 100 steps wide). Complete here your own steps and degrees. Call the teacher.



```

when clicked
  hide
  point in direction 90
  clear
  pen up
  set pen color to blue
  set pen size to 3
  go to x: 0 y: 0
  pen down
  repeat 100
    move 100 steps
    turn 90 degrees
    wait 1 secs
  turn 90 degrees
  repeat 3
    move 100 steps
    turn 90 degrees
    wait 2 secs

```

9 Making quizzes. Make the quizzes below starting with Quiz1 (read the tips below).

Quiz1

```

when clicked
  say This is my first quiz for 2 secs
  ask What's 1,25 X 4? and wait
  if answer = 5 then
    say well done! for 2 secs
  else
    say Sorry! Try again for 2 secs

```

Quiz2

```





when clicked
  say Hi! I am going to calculate for 2 secs
  say how many letters your name has for 2 secs
  ask What's your name? and wait
  set name to answer
  say length of answer for 2 secs
  say join Goodbye name for 2 secs

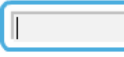

```

- The quiz needs a question to ask the user from Sensing: What is 1,25 X 4?


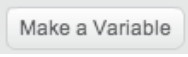
- Depending on whether (if) the answer is correct or incorrect, the quiz has to display (show) a different message: Well done! (correct) or/else Sorry! Try again (incorrect).

- Use a equals block  from  into 



- Place  from  in one side of the  block, followed by  on the other side.

- Run your program, enter your answer in the box  and click on 

➤ Now we are now going to add a score to our quiz using a **variable**. We use a variable to store numbers, text and other characters. Its contents can be changed or varied. We will use the variable to store the score. Follow the steps below:

- Click on  and select . Give the variable the name **Score** and click OK.



- When the quiz starts, the score has to start at zero:  (initialization).

- Every time the user gets an answer correct, we need to increase the score by one  before 




- Run your quiz and check the score starts at zero and it increases by one when a question is answered correctly. Call the **teacher**.

- **Extra point:** make the cat move or dance when an answer is correct. Call the teacher.


10 **Making quizzes.** Now create **Quiz2** (above). First, read the instructions below:

- The sprite is going to calculate the number of letters your name has. So we need to ask for the name  and store the  in a variable.

- Let's create the variable **name**. Click on  and select . Give the variable the name **name** and click OK.



- Now drag the block  to the script area and place  on .

Now the variable  stores the name entered.



- The operator  returns the numeric value of the length of the text string. For example, the length of the string "cu2morrow" is 9, 9 characters.

- Finally, we use the operator `join` `hello` `world` to join (link, connect) 2 strings: "Goodbye" and the name stored in the variable `name`.
- Run the program and call the `teacher_08`.

Scratch: games and stories


11 **Game 1: the paddle.** Create a game in which the player will move a paddle  to keep a bouncing ball  from hitting the floor (based on the arcade game Pong).


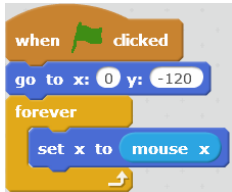
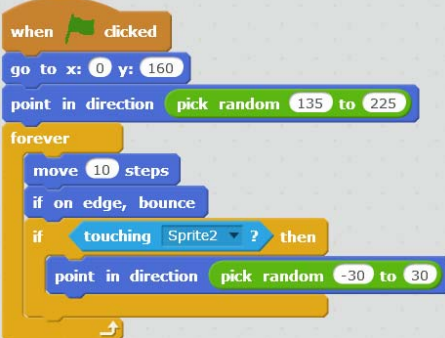
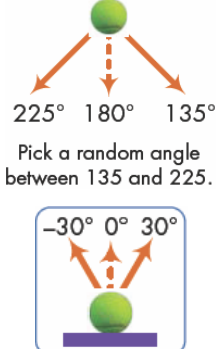
- Draw a new backdrop using Paint:  (floor)

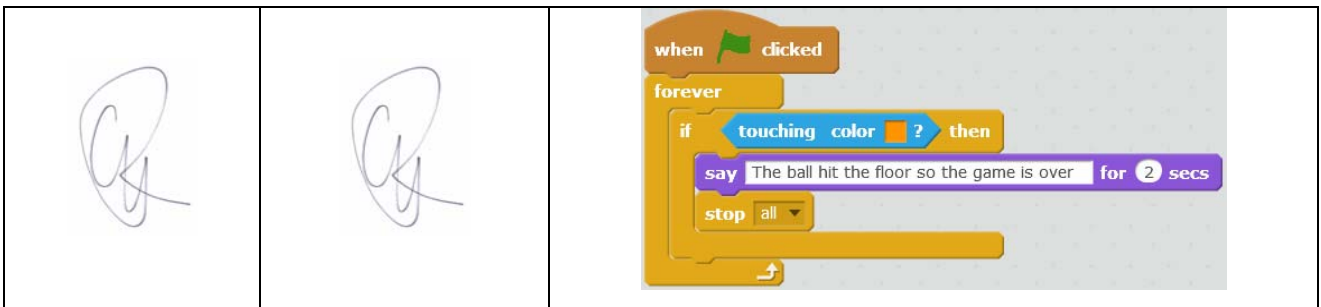
- Create 2 new sprites: a paddle and a ball:  
- Create for each sprite their scripts (the program)

- Sprite 2:** `go to x: 0 y: -120` sets the paddle's vertical position (just above the floor). `set x to mouse x` moves the paddle **back and forth** by matching the paddle's x-position to that of the mouse.

- Sprite 3:** First, we move the ball to the top of the stage `go to x: 0 y: 160` and make it **point down** at a **random angle** `point in direction pick random 135 to 225`. Then the ball moves across the stage and bounce off the edges `if on edge, bounce`. When the ball and paddle touch `if touching Sprite2 ?` the ball points in a random direction (between -30 and 30).

- To finish the game the ball has to touch  (the bottom of the stage) and for that we use `if touching color ?` (we detect the colour).

stage	script for Sprite2	script for Sprite3	
			 <p>Pick a random angle between 135 and 225.</p> <p>-30° 0° 30°</p>



- List 2 changes you can make in the scripts in order to increase the difficulty of the game:

12 **Game 2: guess the number.** Create a game to guess the number the computer thinks (between 1 and 100) in less than 10 attempts.

- Make 2 sprites: the cat (to run the main part) and logo for a presentation
- Each sprite has their own scripts. For the logo only a kind of intro
- For the cat: set variables for the number to guess (**n**), the number of attempts (**att**), the number the player enters (**nread**).
- What do these scripts do?

```
repeat until n = nread or att = 0
```

```
set att to att + 1
```

App Inventor




Day/Date	Signatures		Day/Date	Signatures		Day/Date	Signatures	
1/	Name1:	Name2:	2/	Name1:	Name2:	3/	Name1:	Name2:
4/	Name1:	Name2:	5/	Name1:	Name2:	6/	Name1:	Name2:

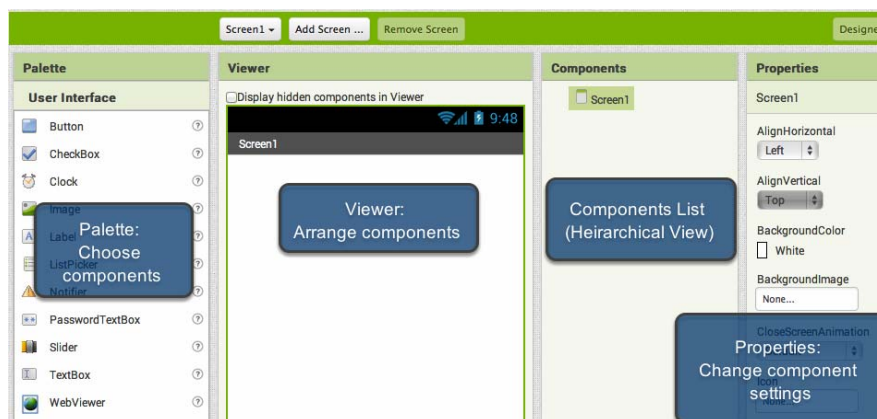
13 **Programming. Understanding App Inventor.** Remember the steps you have to follow to build your App Inventor apps:



<p>Designer is for you to select components for your app and specify their properties.</p>	<p>Blocks Editor is where you assemble program blocks that specify how the components should behave.</p>	<p>Use an Android device to test your app. If you don't have one you can use an emulator (software that runs on your computer and behaves just like the Android device).</p>
---	---	--

➤ To begin with App Inventor, you need a *Google* account. If you do not have one get one at <https://accounts.google.com> (if you're under 13, a parent can sign up for an account at home).

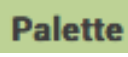

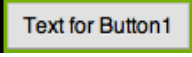
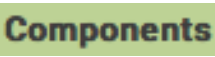
➤ Now go to the App Inventor website , click on  to dismiss the welcome screen (splash screen). Now you are in .





14 **Program01**: talk2me. Create an app that talks to you.


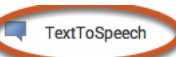



Designing the app screen: buttons, images, text boxes, and functionalities like text-to-speech, sensors, and *GPS*.

•  →  Add a button  (this is your first component ) and see your app on your connected device or the emulator.

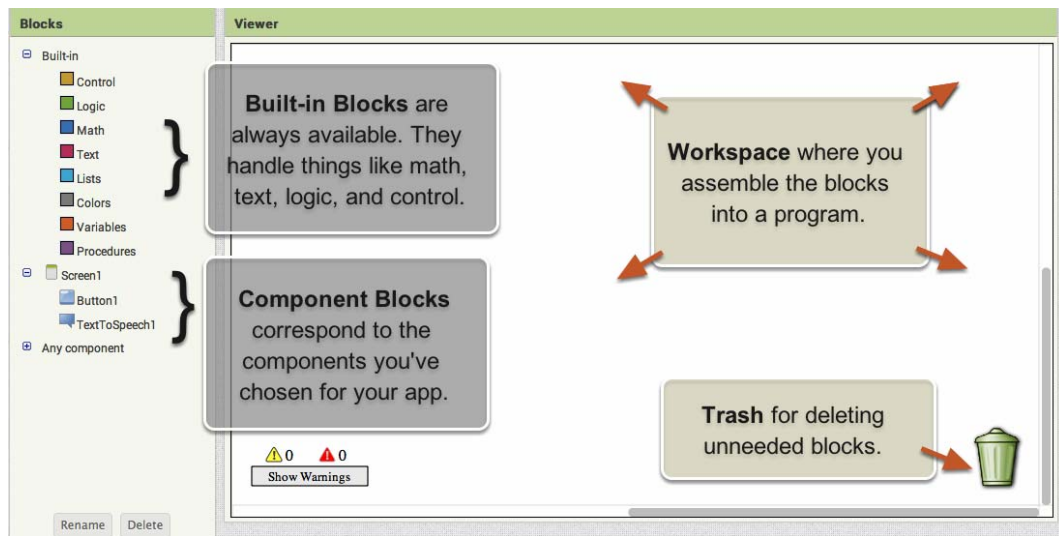
•  Place the button on the centre of the screen (AlignHorizontal: 3, AlignVertical: 2)

•  Change the text on the button to "Talk to Me", shape to oval and the colour to *Blue*.

•   →  Add a text-to-speech component to your app (this component does not show up on the screen but on **Components**).




Telling the app what to do: the Blocks Editor is where you program the behaviour of your app. This editor lets you control how the app works by using programming blocks (similar to Scratch).

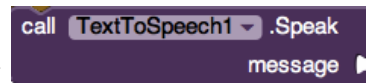




- **Blocks** → **Viewer** Make a button click event. Click on **Button1** to see the options you have for this component. **What we want** is when we click on **Button1** the app speaks.



So select , drag it and drop it on the workspace.

- **Blocks** → **Viewer** Now click on **TextToSpeech1**, select **call TextToSpeech1.Speak message** and drop it over the Button.Click (when you click the button, the app will speak).



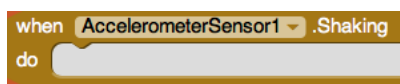
- Now tell the app what to say (message). Click on **Text** and drag a text block  and plug it into **message** . Click on the text box and type (write): "My name is ---- and this is my first app. Hooray"

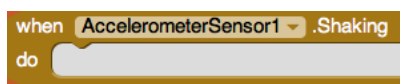


Testing the app. Now go to your connected device (or the emulator) and click the button (make sure your volume is up!). You should hear the phone speak the phrase out loud.


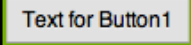



15 **Program02: talk2me2.** Now you will modify your first app so that it **responds to shaking** and lets users put in whatever phrase they want.

- **Palette** → **Sensors** → **AccelerometerSensor** Add an Accelerometer Sensor




- **Blocks** Select , copy (duplicate) the blocks you created for **Button1** (TextToSpeech1 and the message) and plug them into the new block. Now change



the phrase that is spoken when the phone is shaking: "Stop shaking me right now". If you are using a connected device, shake your device and test it out.


- **Palette** → **Viewer** Add a  **TextBox** and put it above 
- **Blocks** Select  (this block makes the app to speak out loud whatever is currently in the TextBox) and drop it on the workspace. Now pull out the text "My name is---" and drag it into the Trash. Plug the TextBox1 into the . When you run the app an empty box will allow you to write a text and if you click on  the text will be spoken out loud.
- So your app has two behaviours: speak out loud the text in the TextBox1 and say "Stop shaking me right now" when the phone is shaken.


16 **Program03**: game01_ballbounce. Create an animation in which a ball bounces around on the screen. **Project** → Start new project.



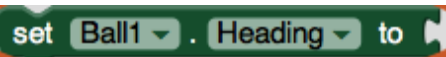
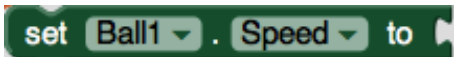
- **Palette** → **Drawing and Animation** →  **Canvas** → **Viewer**
- Adding a **canvas**: the **screen in your app** where you can hold images and animations.
- The screen of your app is **scrollable** (you can **move the cursor up and down** the screen).
- When using a Canvas, we have to **turn off** this property (**Properties**) so that the screen does not scroll. This will allow us to make the Canvas to **fill up** the whole screen:



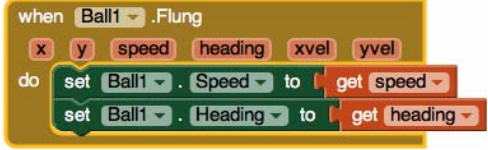
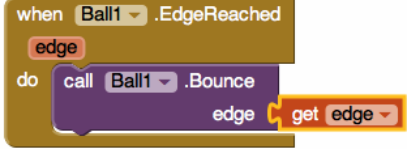
- Changing the height (↕) and width (↔) of the Canvas:  →  and do the same with the Width.

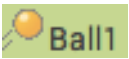
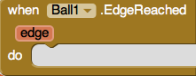
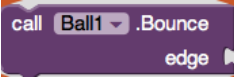
- **Palette** → **Drawing and Animation** →  **Ball** → **Viewer** Adding a ball sprite.

Change its **radius**:  → (**Properties**) **Radius** 15. The app screen is designed. Now let's program the components (the canvas and the ball): **Blocks**.

-  →  to make a **fling gesture** (like when you **throw** something) with the finger. Select  and 

to set (put) the ball's speed and direction. Now place the cursor over **speed** and take **get speed** and plug into the setBall1. Do the same with **get heading**. The final instruction looks this way:

Fling the ball against the edge and...	... the ball will bounce off the edges of the canvas
	



 to make the ball bounce of the edge of the screen. Now select  and place the cursor over **edge** and take **get edge**. The final instruction is in the table above. Now test it out.