

Computing

Lesson 2: Tappy Tap App

Mobile App Development

Ben Garside



Task 1 – Game development

So far, you have:

- Created an app with three screens
- Added an event to a button that will move to the game screen when it is clicked
- Added a timeout to automatically move to the score screen after five seconds

Your code window should look like this:

```
1  onEvent (▼ "startbutton", ▼ "click", function() {  
2    setScreen (▼ "Game");  
3    setTimeout ( function() {  
4      setScreen (▼ "Score");  
5    } , 5000);  
6  });
```

Source: code.org

Work through the steps on the following slides to develop your game further

You can remix this app if you prefer:
oaknat.uk/comp-AppLabL2S



Task 1 part 1 – The blue dot

Task:

Add the blue dot to the game window and give it a sensible id such as “bluedot_game”.

Guidance:

Make sure that you are in Design mode and have selected ‘Game’ from the drop-down menu.



Source: Code.org



Task 1 part 2 – Player score

Task:

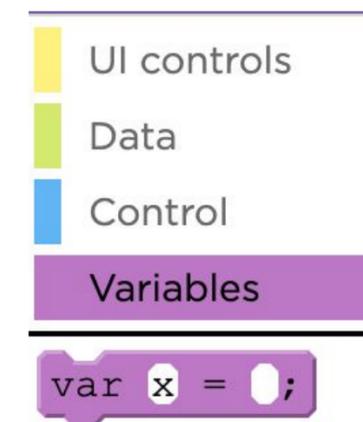
We need a variable so that we can make our program hold and increment a score.

Create a variable and name it 'score', setting the initial value to 0.

Think about where in your program 'score' should be declared and set to 0.

Guidance:

Go to the Variables menu and select the following block:



Source: Code.org

Note: You may hear people refer to creating a variable as 'declaring a variable', as it has the same meaning.



Task 1 part 3 – onEvent

Task:

Add an onEvent into your coding window, below the code that you started with.

Change the properties so that the id is linked to the bluedot_game and the type is 'click'.

Guidance:

```
setScreen (▼ "Score");  
    }  
    , 5000);  
    }  
    );  
onEvent (▼ "bluedot_game", ▼ "click",  
    {  
    }  
    );
```

Source: Code.org



Task 1 part 4 – Increasing the score

Task:

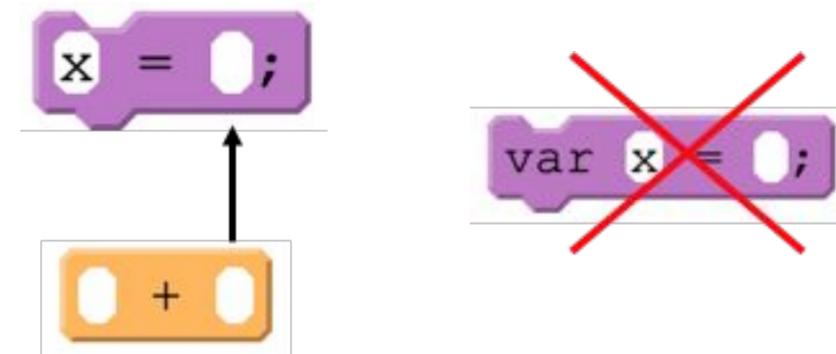
We need to increment the score by one each time the dot has been clicked.

You will need to tell the program to overwrite 'score' with the current value of score plus one:

score = score + 1

Guidance:

To do this, you will need a variable assignment block as below, but will also need to use the Math blocks.



Source: Code.org

Hint: 'x' is the default value in the block. Make sure that you change this to the appropriate variable name.



Task 1 part 5 – Testing part 1

Task:

Try running your game. Does the score increase?

We don't know if our score variable works, as we haven't yet written the code to display the score on the score screen.

In the Variable menu, choose the `console.log(message)` block and place it below where you are incrementing the score.

Try running your program and click the blue dot. You will see the string 'message' appear every time you click the blue dot.

Guidance:

Add the following block of code directly beneath the `score = score + 1` line of code:

```
console.log(message)
```

Source: Code.org

Look at the bottom of your screen and you should see the following each time you click the blue dot:



Task 1 part 6 – Testing part 2

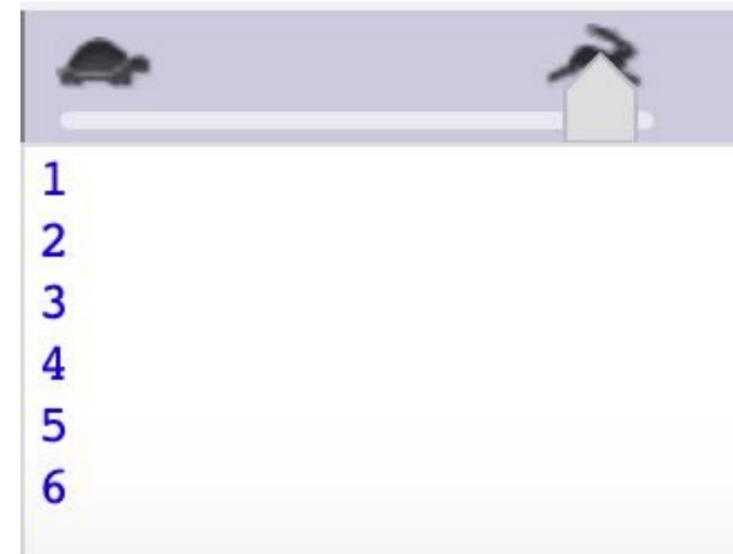
Task:

We now know the onEvent works and that the console.log block of code has been executed.

What can you change about the console.log block so that we can test that the score is being increased each time the blue dot has been clicked?

Guidance:

You will know that you have successfully completed this step if when you run your program, you see numbers incrementing by one each time you click the blue dot:



Source: Code.org



Task 1 part 7 – Move the blue dot

Task:

Our game isn't very fun at the moment, as the dot doesn't move when it is clicked.

We already have an onEvent for what to do when the dot is clicked. Now we need to tell the dot to move after the score has been incremented.

Find the setPosition block from the UI controls menu.

Guidance:

Use the following block and set the id to 'bluedot_game':

```
setPosition(▼ "id", 0, 0, 100, 100); ↵
```

Source: Code.org



Task 1 part 8 – Random movement

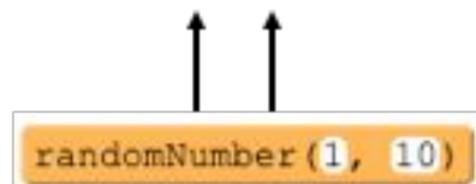
Task:

Your blue dot should have moved, but only the first time that you clicked it. This is because the x and y coordinates were set to 0.

Your screen has x and y coordinates so that you can customise where you would like an object to be placed.

Under the Math menu, find the randomNumber(1,10) block. Replace the 0s for x and y with two of these blocks, as below:

```
setPosition("id", 0, 0, 100, 100);
```



Guidance:

To find the x and y coordinates of the screen, click on the mobile phone screen and move your mouse. You will see the range of x and y coordinates appear and change as you move your mouse around the screen.



Source: Code.org



Optional explorer tasks

Optional tasks:

1. Add a red circle to the game screen using the Design window.
2. Add an onEvent that has the same functionality as the blue dot, the only difference being that the score should decrease by one when it has been clicked.
3. Explore the properties of the setPosition and edit the width and height so that each time the dots are clicked, they also change to a random size.

