SCRATCH TUTORIAL: SKY HUNTER

DIGITAL CONTENT CREATION > 3.4 PROGRAMMING

TARGET GROUP	AGE GROUP	PROFICIENCY LEVEL	FORMAT	COPYRIGHT	LANGUAGE
School drop outs, Students (primary school), Students (secondary school)	Children, Teenagers	Level 2	Activity sheet	Creative Commons (BY-SA)	English, French

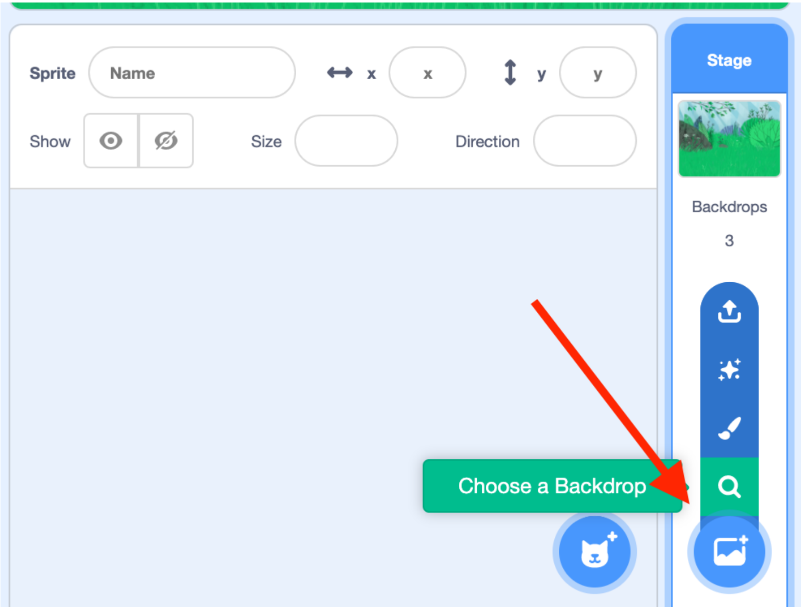
To program this game, we will use the Scratch platform which has the advantage of using visual code: no need to learn a programming language to build your own games and videos!

General Objective	Skillset building
Preparation time for facilitator	less than 1 hour
Competence area	3 - Digital content creation
Time needed to complete activity (for learner)	0 - 1 hour
Resource originally created in	French

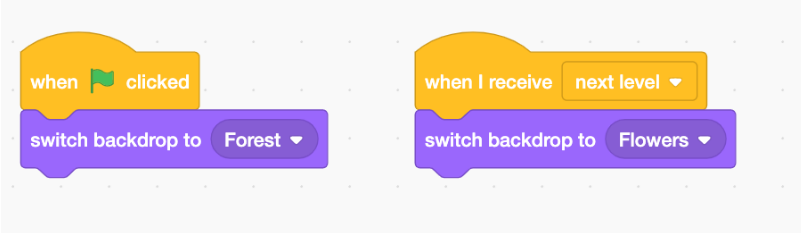
WORKSHOP DIRECTIONS

1 Creating the backdrop

1. We will first need to select a backdrop. Click on the backdrop icon on the bottom right of the creation screen. Select two from the library, from the ‘Outdoors’ category for example.



2. We will now program the two backdrops. Copy the following code into the backdrop’s script by going to the ‘Code’ tab near the top right of the screen:



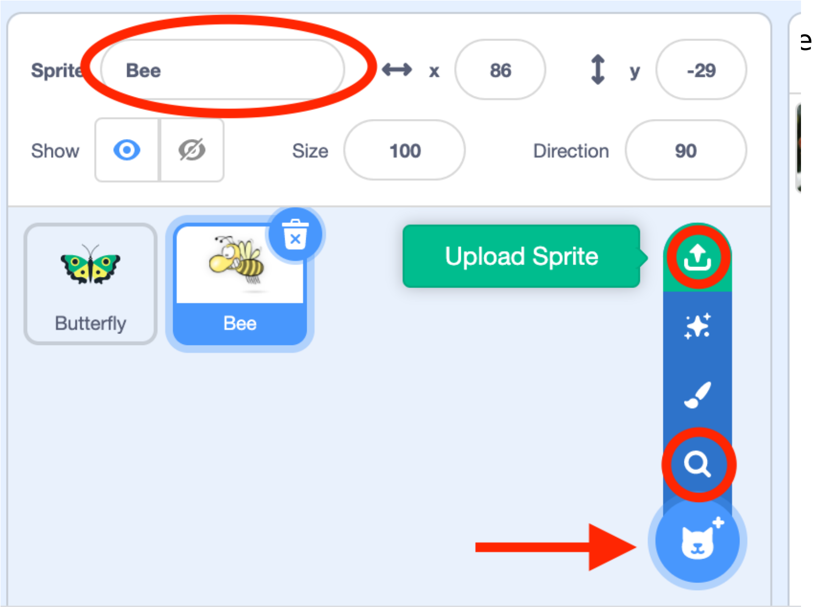
```
when [flag] clicked
switch backdrop to (Forest ▾)

when I receive (next level ▾)
switch backdrop to (Flowers ▾)
```

When the green flag is clicked, the ‘Forest’ backdrop will appear.

2. When the message ‘next backdrop’ is received, the ‘Flowers’ backdrop will appear.

2 Creating the sprites

1. Add the sprite ‘Butterfly 1’ from the library and a ‘bee’ (found here) Click on the sprite icon and then the search icon and upload icon to import from the Scratch library and your hard drive respectively. You

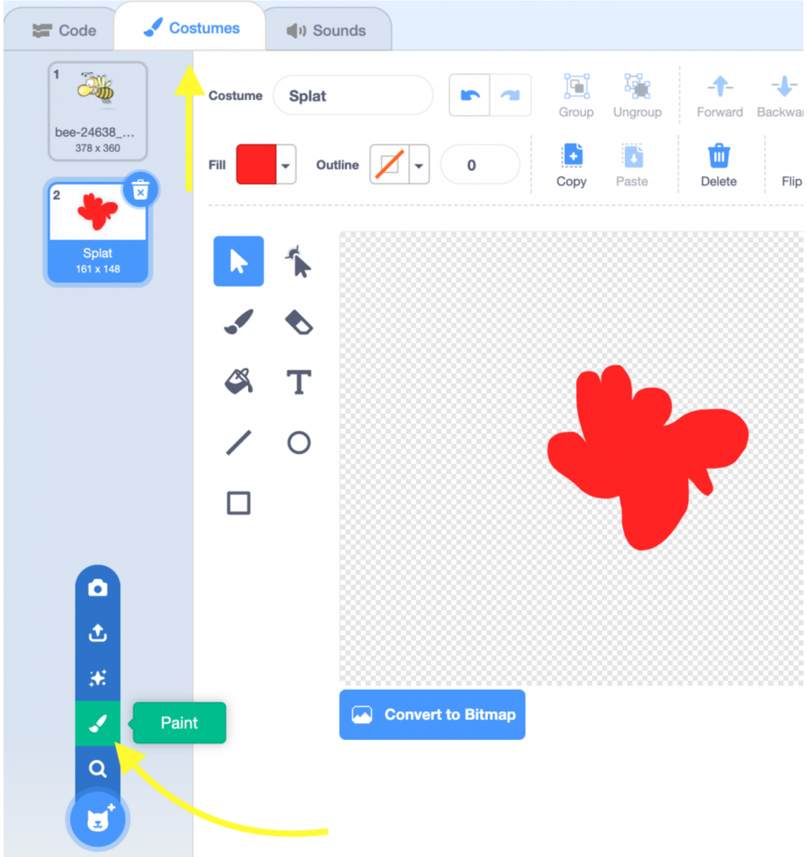


Sprite Bee x 86 y -29

Show Size 100 Direction 90

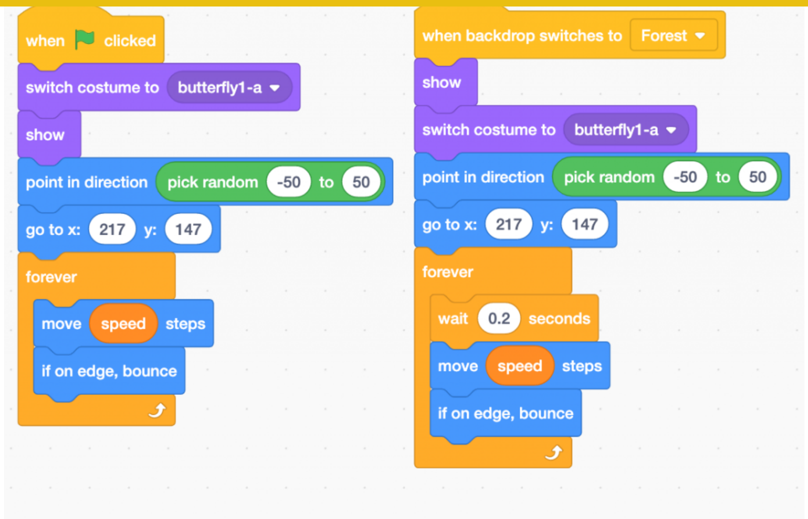
es to ‘Butterfly’ and ‘Bee’ (by clicking on the box

2. We now need to create a second costume for each sprite. We will draw them ourselves by clicking on the paint icon. Draw a kind of ‘splat’ shape symbolising the crushing of the poor insect.



3 Animating the sprites

1. We will need now to create a variable. Click the ‘Code’ tab near the top left of the screen, click ‘Make a variable’ and call it ‘speed’. Tick the box ‘For all sprites’ so that the variable is applied universally, meaning all sprites will move at a defined speed. 2. We will start to program the butterfly’s behaviour. Copy the following to the butterfly’s script:



```
when [flag] clicked
switch costume to (butterfly1-a ▾)
show
point in direction (pick random (-50) to (50))
go to x: (217) y: (147)
forever
    move (speed) steps
    if on edge, bounce
```

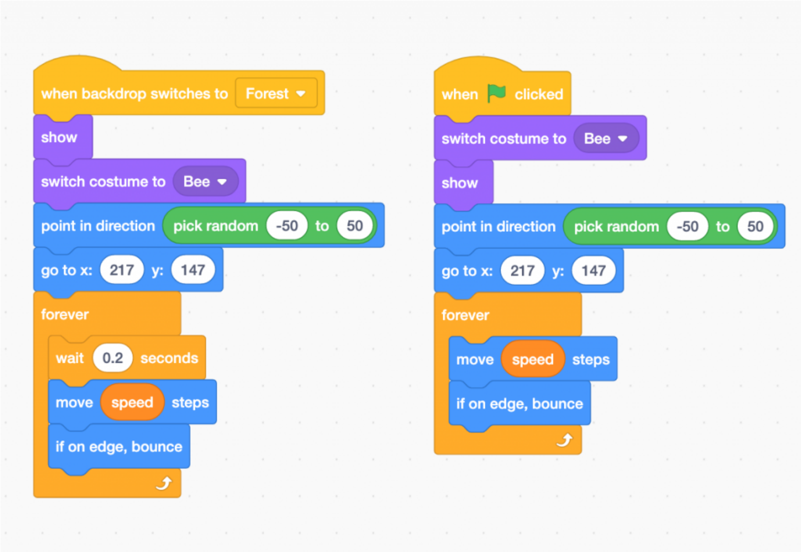
```
when backdrop switches to (Forest ▾)
show
switch costume to (butterfly1-a ▾)
point in direction (pick random (-50) to (50))
go to x: (217) y: (147)
forever
    wait (0.2) seconds
    move (speed) steps
    if on edge, bounce
```

When the green flag is clicked, the butterfly will

appear in its default costume and point in a random direction between -50° and 50°. It will go to the top right of the screen (x = 217; y = 147).

It will then indefinitely move at a speed defined by the variable ‘speed’ until it touches the edge, at which point it will move in another direction. 2. When game changes to the ‘Forest’ backdrop and thus to level 2, the butterfly executes the same actions, the only difference being it waits a number of milliseconds before moving to allow the backdrop to change.

the bee, change the code slightly so that the

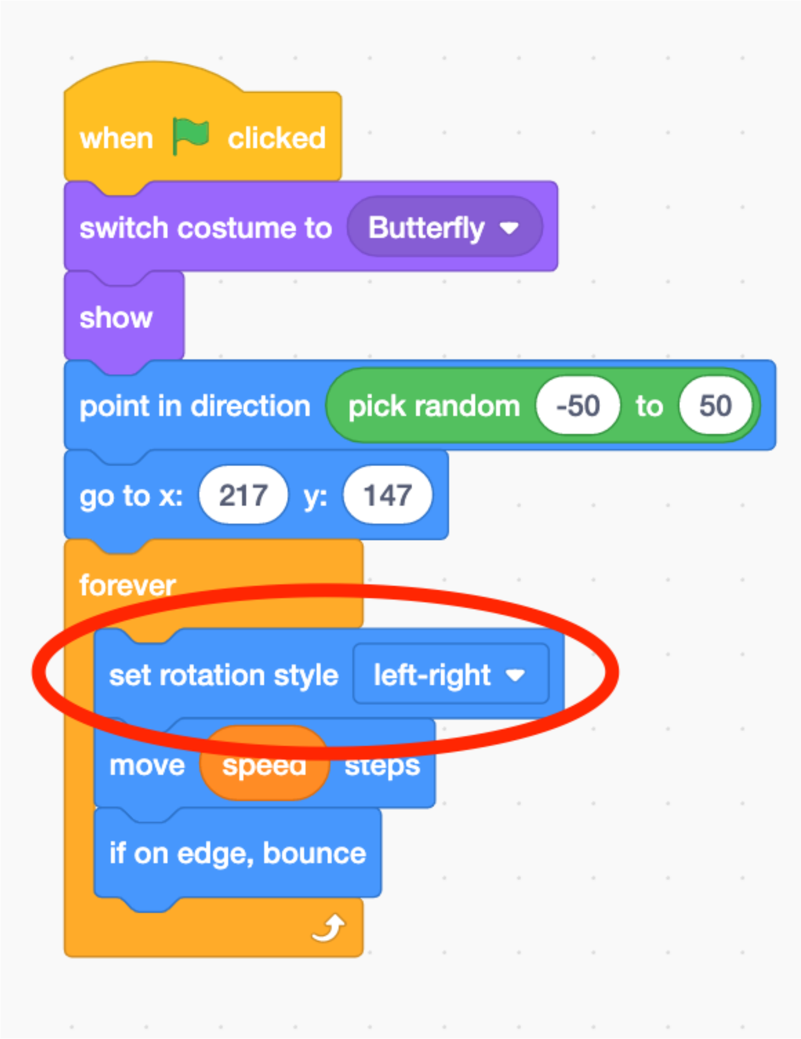


```
when backdrop switches to (Forest ▾)
show
switch costume to (Bee ▾)
point in direction (pick random (-50) to (50))
go to x: (217) y: (147)
forever
    wait (0.2) seconds
    move (speed) steps
    if on edge, bounce
```

```
when [flag] clicked
switch costume to (Bee ▾)
show
point in direction (pick random (-50) to (50))
go to x: (217) y: (147)
forever
    move (speed) steps
    if on edge, bounce
```

The difference in the above is that costume changes to ‘Bee’ (since we’re now in the bee’s script).

4. The sprites can now move in all directions and will sometimes be upside down through their movements. To avoid this, add a ‘set rotation style left-right’ block to each of the block groups you just made. See the circled block below and make sure to add it to each group (i.e. two groups for the butterfly and two for the bee).

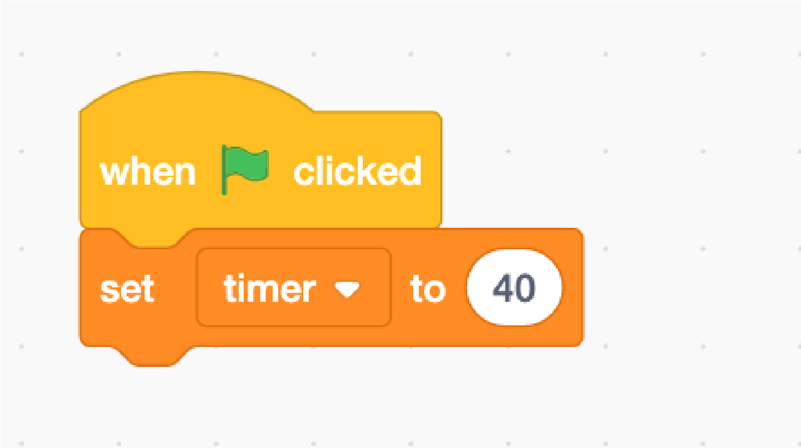


```
when 🏁 clicked
switch costume to Butterfly ▾
show
point in direction (pick random -50 to 50)
go to x: 217 y: 147
forever
    set rotation style left-right ▾
    move (speed) steps
    if on edge, bounce
```

4 Creating a timer

1. We now need to add a timer to the game. Make a new variable, name it ‘timer’ and apply it to all

sprites. Copy the following to the backdrop’s script:

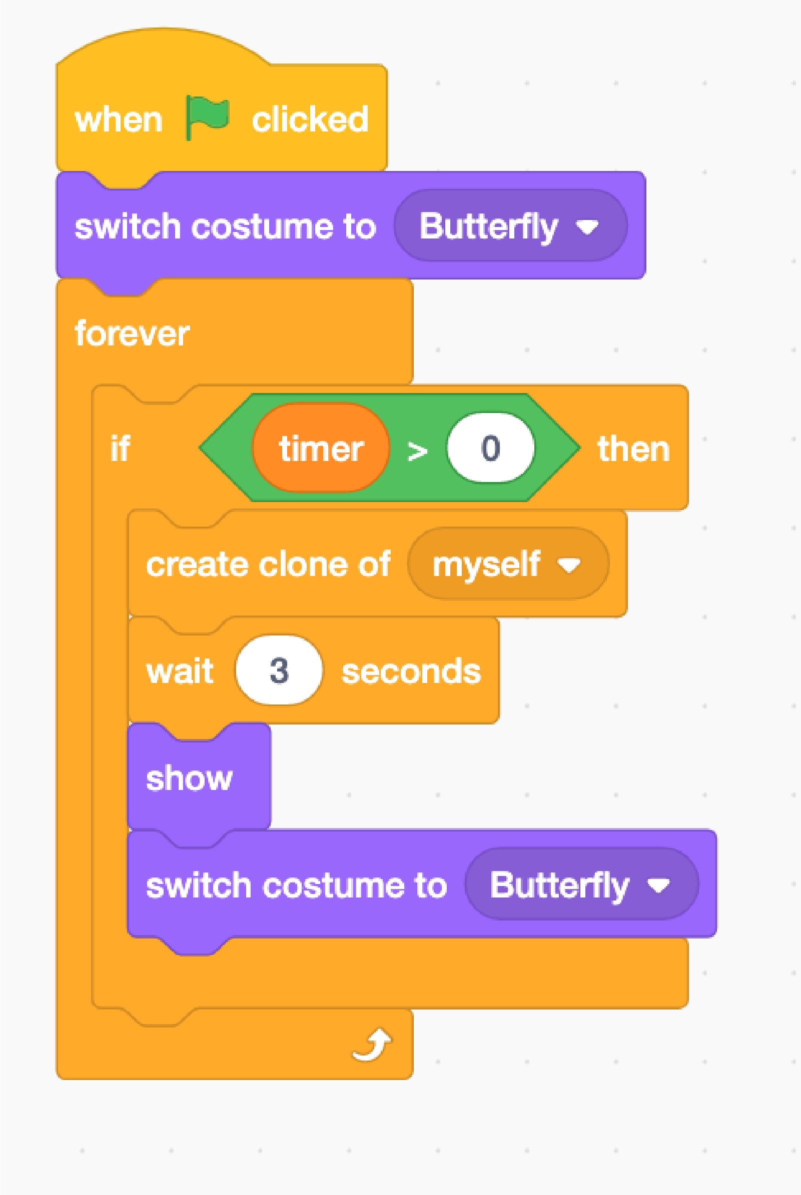


```
when 🟩 clicked
set timer ▾ to 40
```

When the green flag is clicked, the variable ‘timer’ will be set to 40. You can modify the value of ‘timer’ as you wish.

5 Cloning the sprites

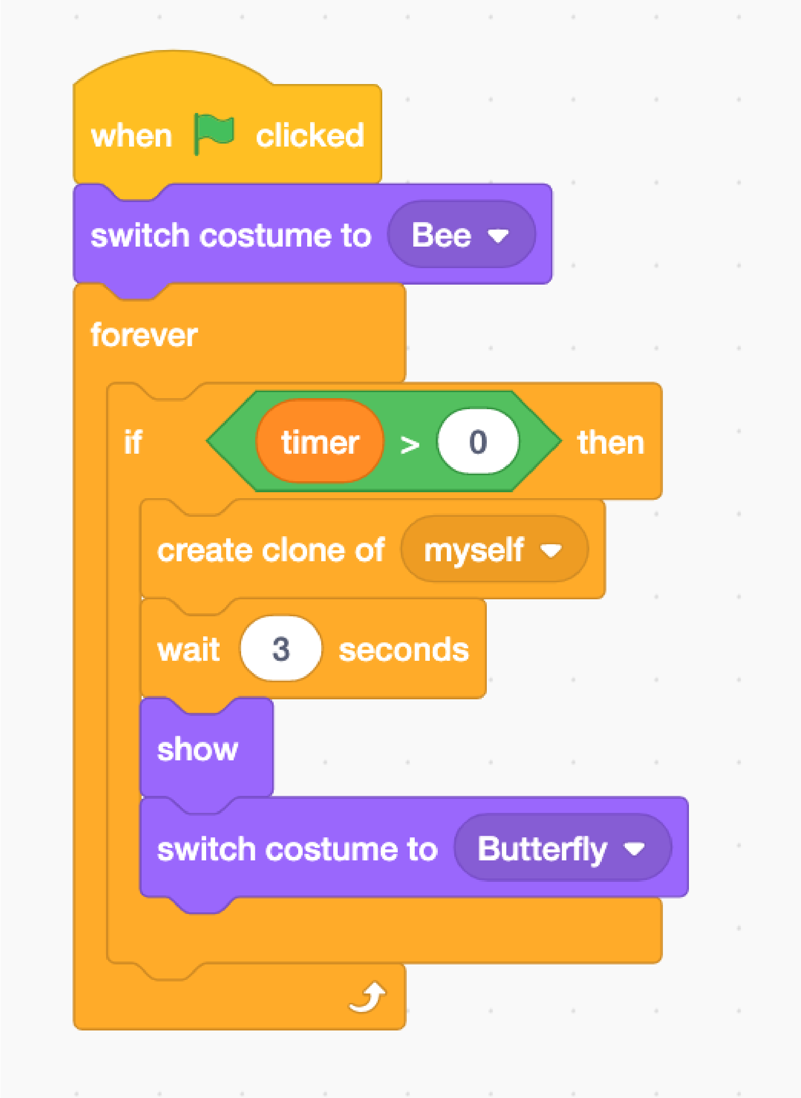
1. To make the game more fun, we will clone the sprites. To begin, write the following code to the butterfly’s script:



```
when 🏴 clicked
switch costume to Butterfly ▾
forever
    if <(timer) > (0)> then
        create clone of myself ▾
        wait (3) seconds
        show
        switch costume to Butterfly ▾
```

When the green flag is clicked, the butterfly changes to its default costume and appears. Now, the following condition is indefinitely verified: if the value of 'timer' is more than 0, the butterfly will create a clone of itself which will then wait 3 seconds, appear in its default costume before starting again (so that the clones will reflect the default butterfly and not the crushed butterfly).

2. Copy this for the bee, not forgetting to change the costume.

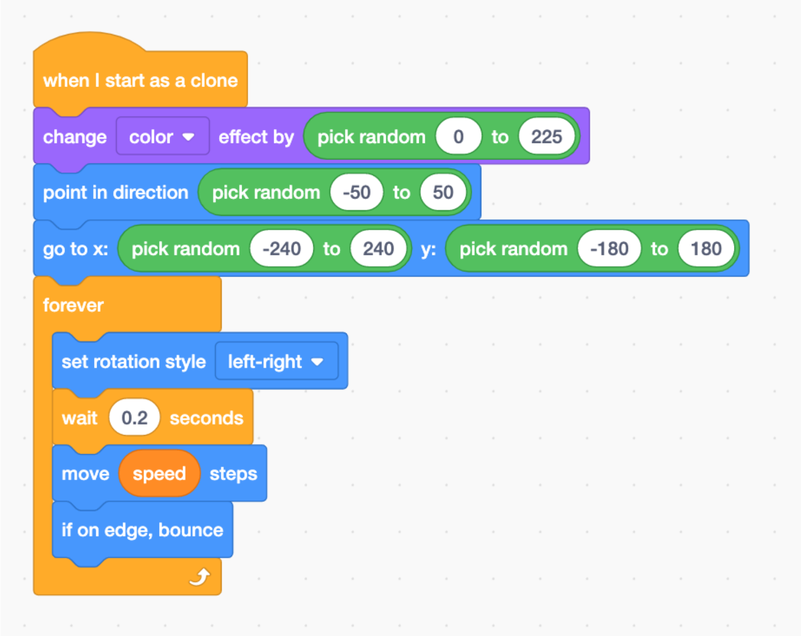


```
when ⚑ clicked
switch costume to (Bee ▾)
forever
    if <(timer) > (0)> then
        create clone of (myself ▾)
        wait (3) seconds
        show
        switch costume to (Butterfly ▾)
    ↺
```

(Note the change in costume from ‘Butterfly’ to ‘Bee’ since we are now in the bee’s script.)

This code means clones will be continually created until the timer reaches 0.

3. To be able to interact with the clone, we have to use the block ‘when I start as a clone’. In your example, we would use the following code:



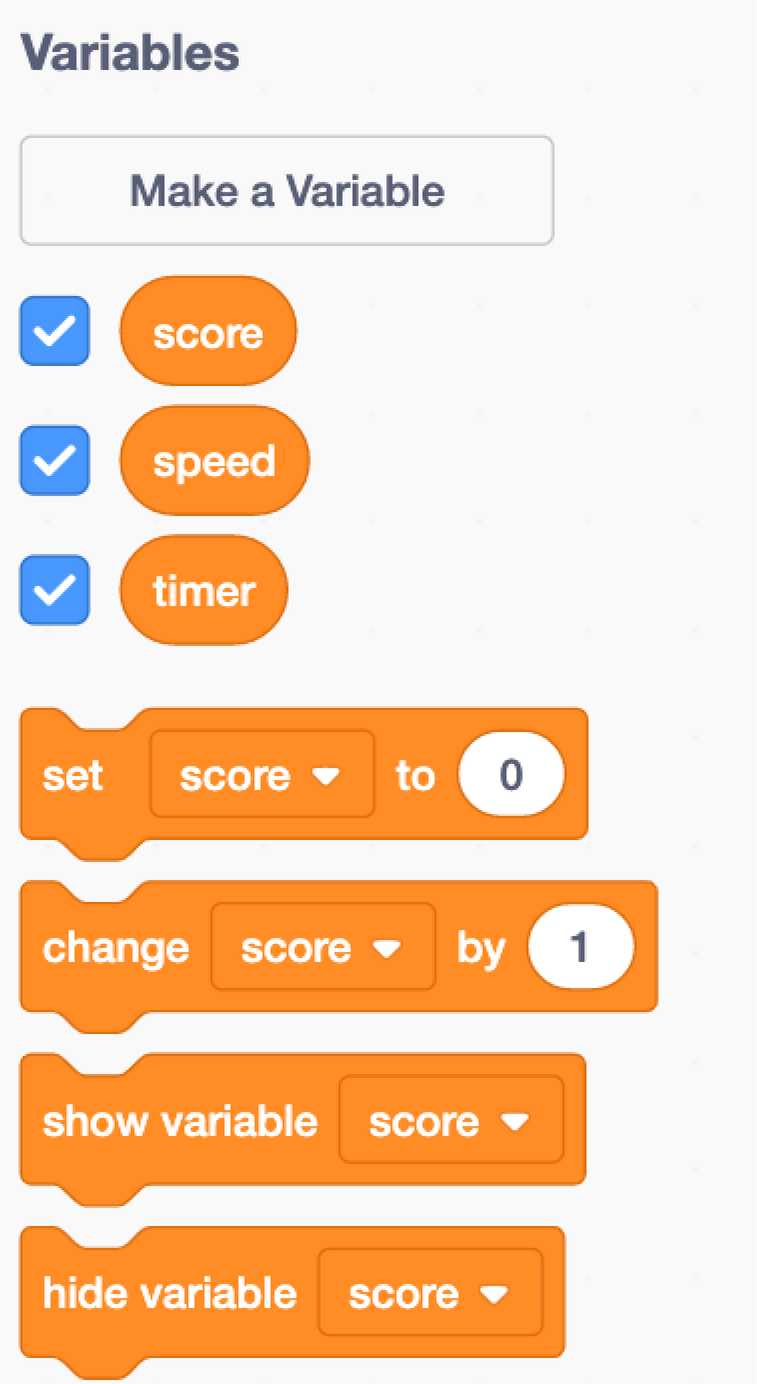
```
when I start as a clone
change color ▾ effect by pick random 0 to 225
point in direction pick random -50 to 50
go to x: pick random -240 to 240 y: pick random -180 to 180
forever
    set rotation style left-right ▾
    wait 0.2 seconds
    move speed steps
    if on edge, bounce
```

When a clone appears – at the start of its program – it changes its colour randomly and points in a random direction starting from a random point. Then, indefinitely, it will wait a couple of milliseconds before moving at a speed defined by the variable ‘speed’. If it touches the edge of the screen, it will rebound, moving in another direction.

Write this into the scripts of the butterfly and the bee. As a result of this code, the sprites will move randomly during the game. We have also added a colour effect so that the clone of each sprite will have a different colouring.

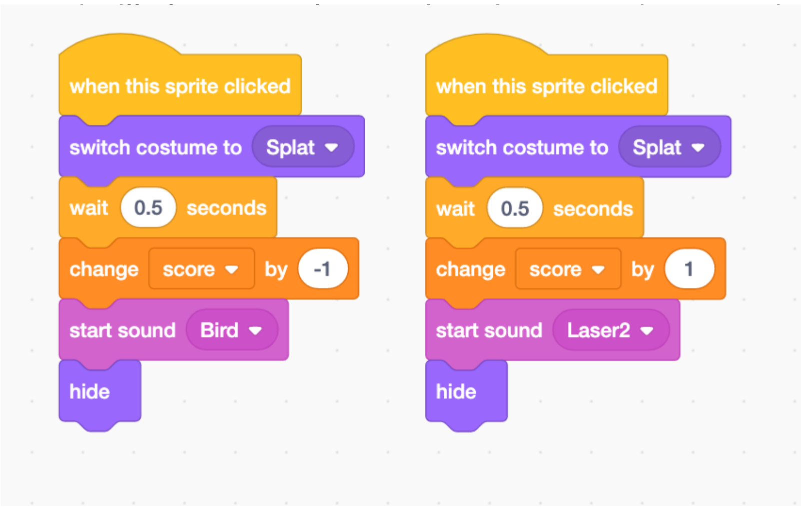
6 Add a point counter

1. Add another variable applied to all sprites, calling it score. We should now have three variables.



2. Now we need to make it so that when we click on the butterfly, we lose a point, while clicking on the bee means we win a point. Write the following two block groups into the scripts of the butterfly and bee respectively. We can add to this to make the game more interesting. We have chosen the ‘bird’ sound

for the butterfly and ‘laser 2’ from the ‘electronic’ category for the bee. When we click on a sprite, its costume. Finally, the sprites need to disappear



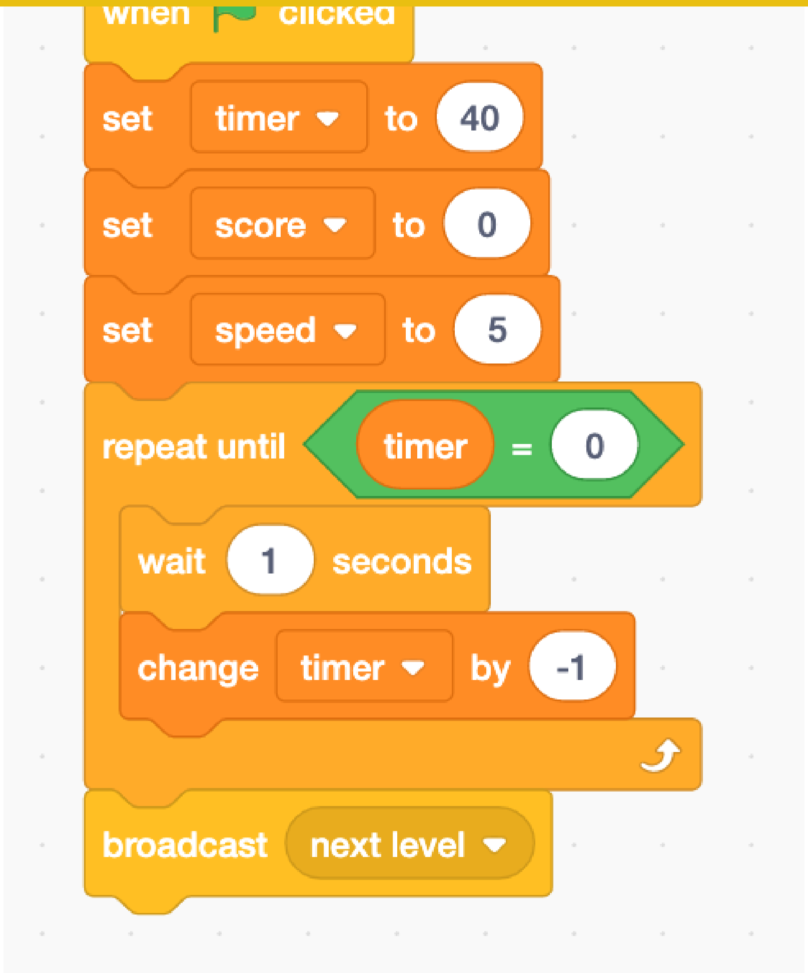
```
when this sprite clicked
switch costume to Splat
wait 0.5 seconds
change score by -1
start sound Bird
hide
```

```
when this sprite clicked
switch costume to Splat
wait 0.5 seconds
change score by 1
start sound Laser2
hide
```

When the player clicks on a sprite, it will change to the ‘splat’ costume, wait a half second before removing 1 from the variable ‘point’ (if it’s the butterfly) or adding 1 instead (if it’s the bee) and play a sound (‘bird’ for the butterfly or ‘laser 2’ for the bee). Finally, they will disappear.

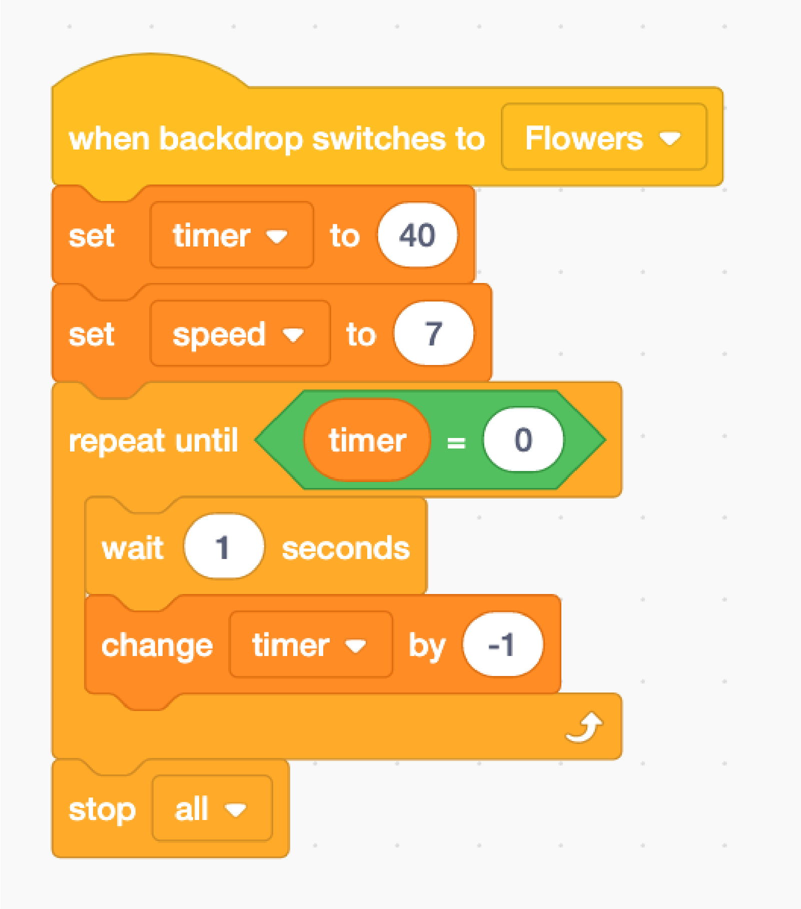
7 Synchronising the parts

Finally, we will put all the elements together. Click on ‘Stage’ on the bottom right of the screen, and then on the ‘Code’ tab near the top left. Copy the following blocks to the backdrop’s script.   1. For level 1, write the following code:



```
when [flag] clicked
set timer to 40
set score to 0
set speed to 5
repeat until timer = 0
    wait 1 seconds
    change timer by -1
broadcast next level
```

When the green flag is clicked, the variables are set at their defined values. Then, the ‘timer’ variable will lose 1 every second until it is equal to 0. From then on, the ‘next level’ message is sent which will allow the program to change the relevant elements. 2. Now for



```
when backdrop switches to [Flowers ▾]
set [timer ▾] to (40)
set [speed ▾] to (7)
repeat until <(timer) = (0)>
    wait (1) seconds
    change [timer ▾] by (-1)
end
stop [all ▾]
```

level 2:

When the backdrop changes, the variables adopt these new defined values. Then, as before, the ‘timer’ variable will lose 1 every second until it is equal to 0. At that point, the program stops. We used 5 for the speed of level 1 and 7 for level 2. This will allow you to ease into the game and add difficulty in the second part. We have also implemented a countdown which will stop when the timer reaches 0.

8 Bonus

Ways to improve the game :

- Make more levels
- Add more creatures
- Add bonuses and/or penalties
- Increase the speed

- Make pick-ups (which could for example add seconds to the timer) or objects to be avoided (which if clicked could for example make more clones appear)