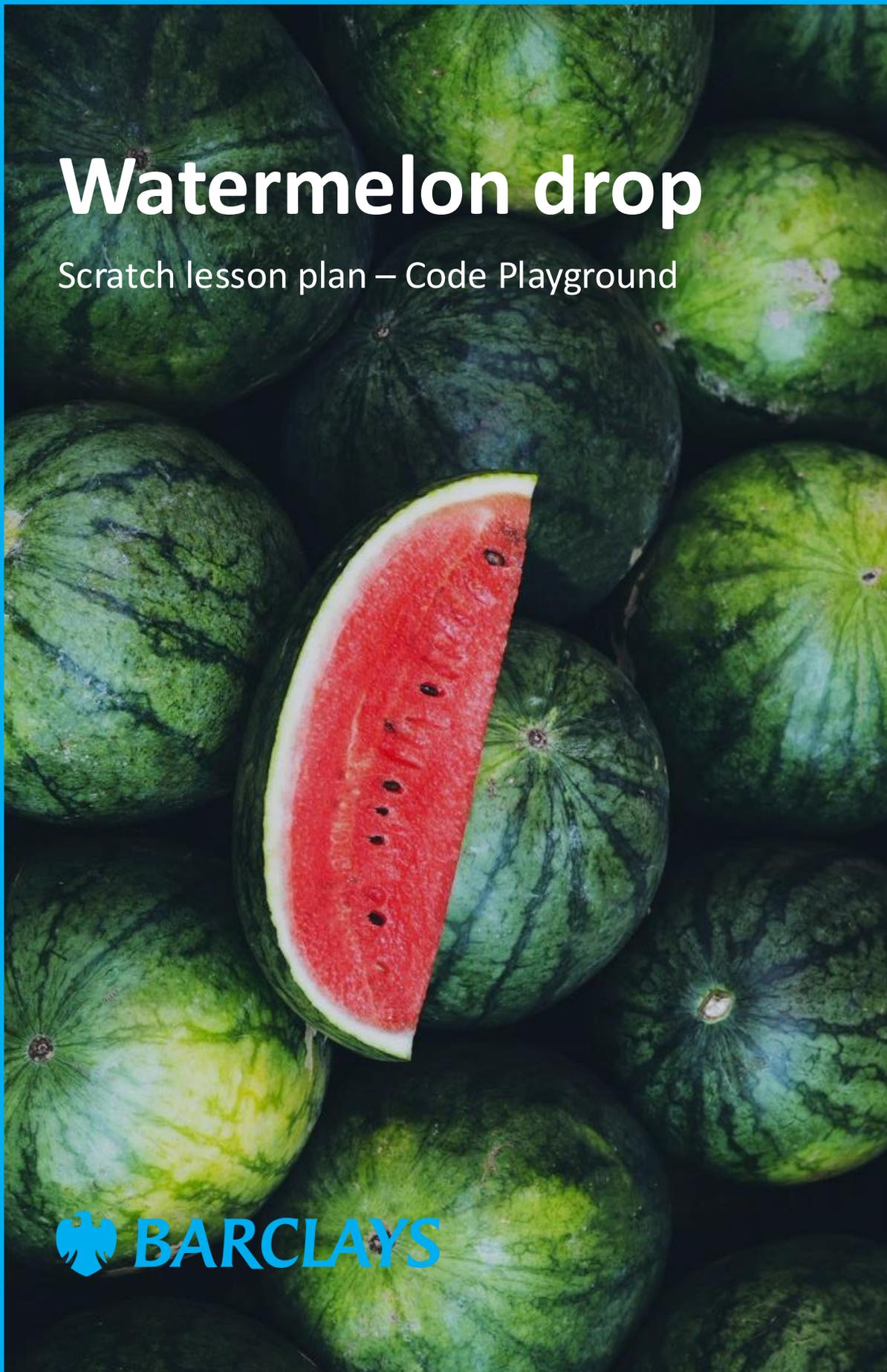




Watermelon drop

Scratch lesson plan – Code Playground



Lesson overview

In this lesson, students will create a Scratch game called Watermelon drop, where a monkey throws watermelons at a lion. They will learn to program movement, use costume changes for animations, and create interactive gameplay using loops, events, and sensing blocks. This lesson uses the Watermelon drop workbook and instructional video to guide the class step-by-step.

Time	Key learning outcomes	Resources
45 mins	<ul style="list-style-type: none">• Use Scratch to program sprite movement with arrow keys• Create animations using costume changes• Program interactivity through sensing blocks and broadcast messages• Apply loops to repeat actions efficiently.	<ul style="list-style-type: none">• Laptops or desktop computers• Access to Scratch website - https://scratch.mit.edu• Watermelon drop workbook for students• Watermelon drop video for teacher reference.

Content

Activities	Time	Page
Lesson introduction	5 mins	3
Activity – Watermelon drop	40 mins	4
Code snippets	-	6
Summary	-	7

Lesson introduction (5 minutes)

Today, we're creating a game called Watermelon drop, where a monkey throws watermelons at a lion. This will teach us how to animate characters, control their movement, and make a fun, interactive game.

Show the class the demo of the completed game from the video. Pause to highlight key features, like how the monkey moves, the watermelon is thrown, and the lion reacts when hit.

We'll use loops to repeat actions, sensing blocks to detect when sprites touch, and messages to control the game's flow.

Games like this teach us the basics of programming animations and interactions—skills used in creating video games and apps.



Activity – build the game (40 minutes)

Step 1 – Choose a backdrop (5 minutes)

1. Guide students to choose the '**Jungle**' backdrop by clicking '**Choose a backdrop**' and selecting it from the library
2. Ensure students understand the role of the backdrop – how it sets the scene for the game.

Step 2 – Add the main character (5 minutes)

1. Help the students to add the '**Monkey**' sprite by selecting the sprite from the Scratch library and resizing it to fit the scene
2. Have them move the monkey into place at the top near the branches, ready to throw the watermelons.

Ask the class

- Why is it important to position and size sprites before coding?

Step 3 – Make the monkey move (5 minutes)

1. Follow the workbook instructions to program left and right movement:
 - Add 'when [right arrow] key pressed' and 'change x by [10]
 - Add 'when [left arrow] key pressed' and 'change x by -10'
2. Ask students to test the monkey's movement.

Ask the class

- What happens if you increase the value in the movement block? How would it change the monkey's speed?

Step 4 – Make the monkey throw (5 minutes)

Program the monkey's throwing animation by adding '**switch costume to [monkey b]**' for throwing effect. Followed by a '**wait [1] seconds**' then switch back to '**monkey a**' costume.

Explain how costume changes create animations in Scratch. Then have them press the spacebar to ensure the animation throws correctly.

Activity – continued

Step 5 – Add the watermelon (5 minutes)

1. Add the '**Watermelon**' sprite and resize it to fit in the monkey's hand
2. Program the watermelon to '**hide**' at the start by adding '**when  clicked**' and '**hide**' as a block.

Step 6 – Throw the watermelon (10 minutes)

Program the watermelon to move from the monkey to the bottom of the screen

- Use '**go to [Monkey]**' and '**show**' to position it correctly
- Add a loop with '**change y by [-10]**' to make it fall

Test the movement of the watermelon, does all the classes stop at the bottom of the screen?

Ask the class

- What happens if you increase the value in the 'change y' block? Does it make it fall faster or slower?

Step 7 – Add the target (10 minutes)

1. Add the '**Lion**' sprite and position it at the bottom of the screen
2. Add '**move [10] steps**' and '**if on edge, bounce**' in a loop to make it move back and forth across the screen
3. The lion will need an extra block adding to keep it from flipping upside down.

Step 8 – Hitting the target (5 minutes)

Program the lion to react when hit by a watermelon, adding '**say**' blocks to show as hit. Then they'll add a reset position to keep the game going.

Throw the watermelon at the lion to check this works.

Step 9 – Making the watermelon disappear (10 minutes)

The last step will remove the starting block for the watermelon and have them replace it with a script that effects whether the watermelon shows or hides when there is contact with the lion.

The workbook will take them through what to add.

Code snippet - examples

Monkey

```

when right arrow key pressed
change x by 10
    
```

```

when left arrow key pressed
change x by -10
    
```

```

when space key pressed
switch costume to monkey-a
wait 1 seconds
switch costume to monkey-b
    
```

Watermelon

```

when space key pressed
go to Monkey
show
repeat until touching edge
change y by -10
hide
    
```

```

when I receive Start
hide
forever
if touching Lion then
hide
go to Monkey
    
```

Lion

```

when clicked
set rotation style left-right
show
broadcast Start
forever
next costume
move 10 steps
wait 0.25 seconds
if on edge, bounce
    
```

```

when I receive Start
forever
if touching Watermelon then
say Roar! You got me for 1 seconds
hide
set x to pick random -180 to 180
wait 1 seconds
show
    
```

Summary

The following information is an example of what a child at an expected level would be able to demonstrate when completing these activities with additional examples to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment guidance

Differentiation – Lower ability/ASN

- Focus on programming the monkey's movement and the watermelon's throw
- Provide support using the code snippets in this plan
- Pair lower ability students with other children for support.

Differentiation – Higher ability/extension

- Challenge students to add advanced mechanics, like power ups or a second level
- Encourage them to use the Pen extension to draw patterns as the watermelon moves.

Plenary

- Show an understanding that different code types can be combined into one project to increase the project complexity
- Understand how collision detection can be used as a method within coding.

Assessment questions

- How did you program the monkey to throw the watermelon?
- What coding blocks were used to make the lion move and react?
- What feature would you add to make your game more interactive?

Extension ideas

- Add a timer to see how many points can be scored in a set time
- Introduce a 'Game over' condition when the watermelon misses too many times
- Add more targets, like a zebra or tiger, with different behaviours.