

# Lesson 5.4: Popping Balloons - Part 1

## Objectives

In this lesson, students will:

- ❖ Experience building a program by experimenting and iterating
- ❖ Gain more familiarity with the X / Y coordinate plane , random numbers and events

## Agenda

1. Introduction	5 mins
2. Flying Balloons	10 mins
3. Student Activity - Flying Balloons	15 mins
4. Moving the Pencil	10 mins
5. Wrap up and Reflections	10 mins

## Preparation

- Remix the game so that you can demonstrate it to students. Become familiar with the code as you will explain it to students step by step.
- Print student activity worksheet (one per student pair)

## Resources & Links

- Link to the **demo** project: <https://scratch.mit.edu/projects/381917240/>
- Link to solution project for **Part 1**: <https://scratch.mit.edu/projects/259880249/>

## 1. Introduction

Students will code a balloon popping game during the student activity. Students will iteratively add additional features to the game over the course of three lessons. In this lesson students will code most of the basic game.

You will demonstrate and explain code sections of the game as a class activity. Students will then engage in activities to code the sections in their own project.



**Demonstrate** the final balloon game and explain the activity so students know what they will be creating: <https://scratch.mit.edu/projects/381917240/>

The pencil is used to pop the balloons; however you can encourage students to use any sprite they like to pop the balloons.

The solution to what students will be coding in this lesson (part 1):

<https://scratch.mit.edu/projects/259880249/>

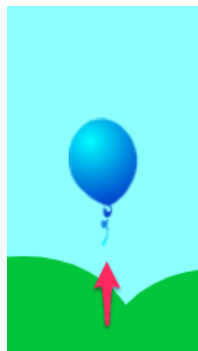
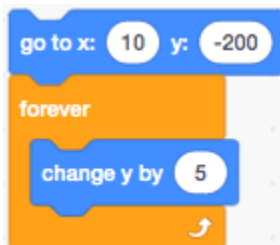
## 2. Flying Balloons



**Display your screen and engage** students in a demonstration and instruction of coding the balloon script:

Create a balloon sprite and explain that we want the balloon to start at the bottom of the stage and slowly move up so that it looks like it is flying.

**Explain** the following code line by line (available in demo project) which is the basic code to make the balloon move up. **The key** is to explain that by increasing the y value in a loop, the balloon's position slowly moves higher and higher as the y value becomes larger.



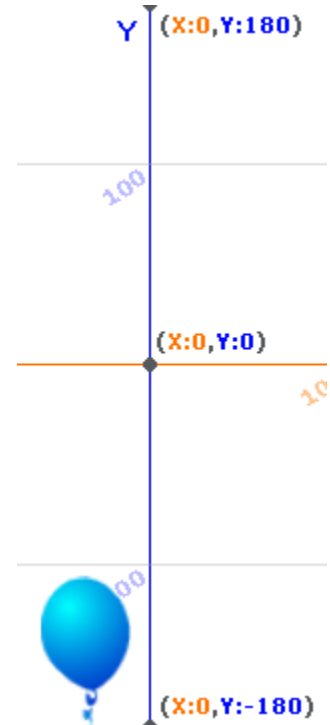
**Demonstrate the code and prompt** students to call out the next desired behaviour, namely to have the balloon fly up again when it reaches the top.

In the game, it appears that a new balloon is flying up once the first balloon reaches the top. To code this, we want to detect when the balloon reaches the top of the screen and then start at the bottom again and then slowly move up again.

**Note:** You can use the grid backdrop in the demo project to help facilitate the position of the balloon on the stage.

What is the position of y when the balloon is at the top? It is a value > 180. To check if the y position is greater than 180, we need both an if statement (a conditional statement explained in the last game) and an operator.

In this case the operator is the > (greater than). What are other programming operators? (multiplication \*, addition +, subtraction -,....)



**Explore** the *operators* category and explore and explain a couple of operational blocks. Then choose which operator will allow us to check when something is greater than.

```

when green flag clicked
  forever loop
    change y by 5
    if y position > 180 then
      go to x: 10 y: -180
  
```

Add the instructions to test if the balloon is at the top of the stage. If it is, the balloon should start back at the bottom before slowly moving up again. Explain and test the code to the left.

Notice how the balloon always flies up from the same position at the bottom? If the balloon always starts flying from the same place, the game would not be very challenging. We want the balloon to start flying at a random location.

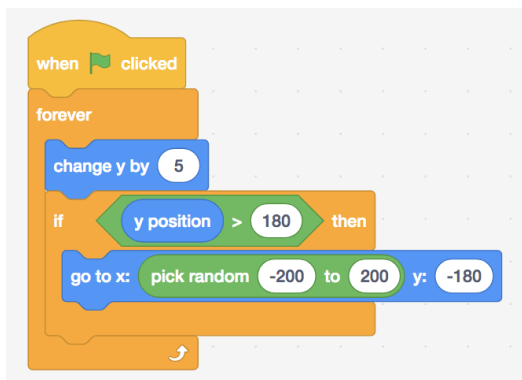
Explain to students that during the student activity, they will add the code to have the balloon start flying from a random position.

### 3. Student Activity: Flying Balloons



Distribute the student activity worksheet and explain the activity to students.

#### Solution Code:



```

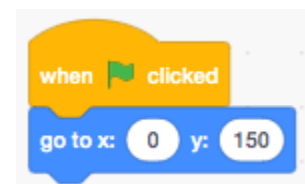
when clicked
  forever
    change y by 5
    if y position > 180 then
      go to x: pick random -200 to 200 y: -180
  
```

### 4. Moving the Pencil



**Demonstrate** how to code the pencil movement.

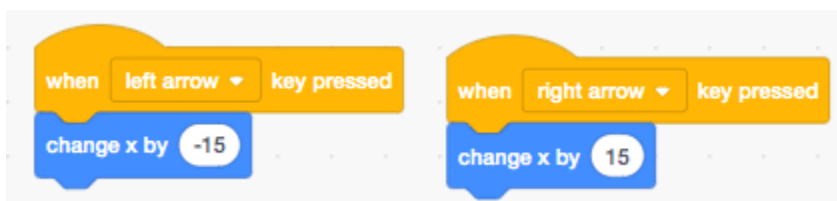
Create a pencil sprite, shrink it in size (Size 45) and initialize its position to be at the top of the screen when the green flag is clicked.



```

when clicked
  go to x: 0 y: 150
  
```

Next, add the code to move the pencil left when the left arrow is clicked and right when the right arrow is clicked. Remind students that the pencil moves are based on the **event** of the left and right arrow keys being clicked. The pencil scripts are very similar to how the diver in the previous game behaved.



```

when left arrow key pressed
  change x by -15

when right arrow key pressed
  change x by 15
  
```

## Students Turn:



Instructions to give to students:

1. Create a pencil or some other sprite to pop balloons
2. Make the pencil smaller by setting the size to something much less than 100.
3. Initialize the pencil's starting position to be at the top of the stage (0, 150)
4. Add an event **when the left arrow key is pressed**. Move the pencil 15 steps to the left (negative number) when the left arrow key is pressed.
5. Do the same for the right arrow, but move the pencil to the right instead.

## 5. Wrap Up and Reflections



During the next lesson, we will add the code to pop the balloon when the pencil touches the balloon and many other cool features for the game.



### Reflection Points:

- What did you learn during this lesson?
- How do we know the balloon is at the top of the stage?
- What was challenging about today's activity?
- To move a sprite from right to left, do we change the **y** or **x position**?

## Student Activity Worksheet: Flying Balloons

What to do:	Using/Details:
<p>Create a new project and give it a name.</p> <p>Create the balloon sprite and delete the cat.</p>	
<p>As long as the program is running, change the position of y by 5.</p> <p>If the balloon reaches the top of the stage (y position is &gt; 180), start over by going to the bottom of the stage.</p>	
<p>Test and experiment until your code works.</p>	
<p>Explore changing the code so the balloon starts flying at a random position between left and right.</p> <p>The stage is about 400 steps wide, 200 to the left from the middle and 200 to the right from the middle.</p>	