Lesson 5.2: Watch Out Diver

Objectives

In this lesson, students will:

- ❖ Learn what a conditional statement is and how to use it.
- ❖ Students are introduced to various new concepts and coding blocks in Scratch such as the forever loop, conditional statement, sensing block, rotation style and the say block.

Preparation

- ❑ Computers with internet connection
- ❑ Remix the Diver game so that you demonstrate it to students. Become familiar with the code as you will explain it to students step by step.

Agenda

1. Diver Game Review	10 mins
2. Making the Fish Move	15 mins
3. Watch Out for Fish	15 mins
4. Wrap Up and Reflections	10 mins

Resources & Links

- ❑ Diver game solution: https://scratch.mit.edu/projects/261565076/

1. Diver Game Review

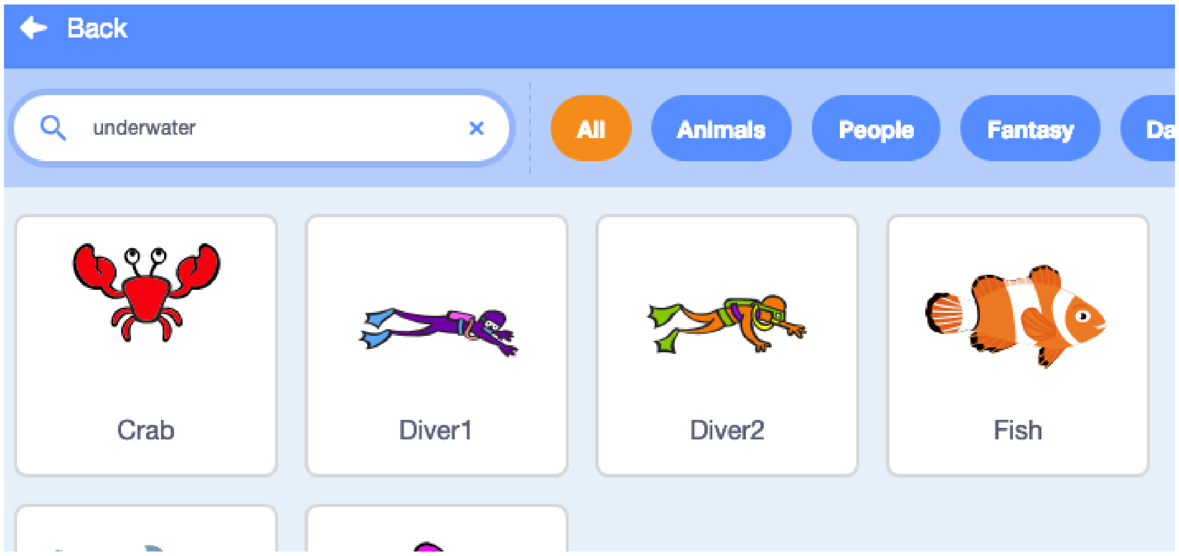
It might be helpful to demonstrate the diver game to students again so they can better continue coding the game.

You will continue to code and explain sections of the game as a class activity followed by student activities in which they code the sections in their own project.

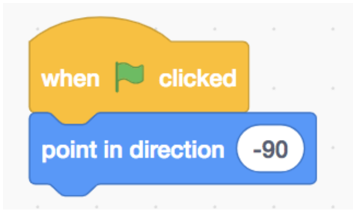
2. Making the Fish Move

Display your screen and engage students in your demonstration and instruction of coding the fish script:

Create the fish sprite. The underwater category helps finding the fish quickly. Show the various costumes for the fish sprite and choose one for the game.

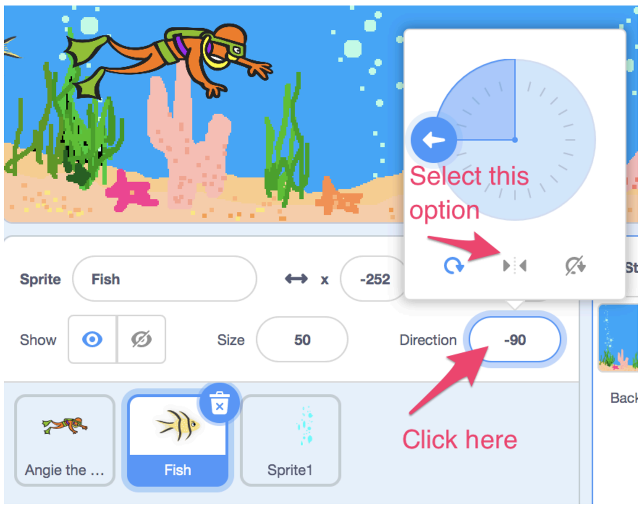


Notice the fish is pointing in the wrong direction because we want the fish to move from right to left towards the diver. To make sure the fish points to the left, add the 'point in direction -90 (left) block from the motion category. We want to do this when the green flag is clicked.



```
when 🏳 clicked
point in direction (-90)
```

Test the code. Notice the fish is pointing to the left, but it is upside down.



To fix this, select the fish sprite and then click on Direction.

A small window will pop up. Select the >< option so the fish turns right to left and not upside down.

Students' Turn:

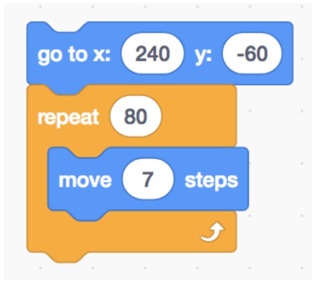
Students open their diver game project and add the fish sprite, select a costume and select the options so the fish is pointing in the correct direction and is right side up.

Continue your demonstration:

Did you notice that in the game, the fish moves from right to left towards the diver and then goes back to the very right of the stage ?

Prompt students for ideas of how to move the fish slowly to the left towards the diver.

To move the diver up and down we changed the value of 'Y'. To move the fish from right to left, we change the value of 'X'. The fish should move slowly a little at a time until it reaches the other side of the stage. To do this we can use the move block inside a repeat block.

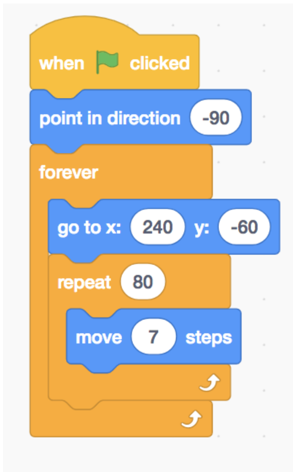


```
go to x: 240 y: -60
repeat 80
    move 7 steps
```

Remember, we need to make sure the fish starts at the right side of the stage at about the diver's height. The x/y values can be examined in the lower right section of the stage as we move the cursor around to find the location where we want the fish to start moving.

Click on the code to test it. **Encourage** students to write some code and test it frequently to see if it works. It is much easier to debug a small amount of code.

The above code moves the fish across the screen once, but we want to do it as long as the program runs. What control statement can we use for that? We use the forever block around the code to move the fish.



```
when [flag] clicked
point in direction (-90)
forever
    go to x: (240) y: (-60)
    repeat (80)
        move (7) steps
```

Let's put it all together now.

Students' Turn:

Students open their diver game project and code the instructions to move the fish across the stage towards the diver for as long as the program runs. You may need to repeat the main sections of the code they will be adding:

1) Add a forever loop to move the fish as long as the program is running
2) Add a go to block to make the fish go to the left side of the stage at the height of the diver
3) In a repeat loop, move the fish slowly across the stage.

3. Watch Out for the Fish ! (Conditional Statements)

Engage students in an interactive class discussion and instruction:

Whenever the fish runs into the diver, the diver says: "Watch out fish !". Before the diver says something, the diver needs to check if the fish touched it. In computer science, when we ask if something is true in order to do something it is called a ***conditional statement***.

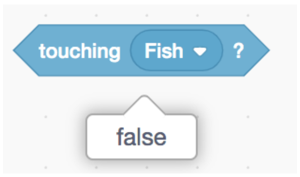
We use conditional statements in real life all the time. For example, "If I'm hungry, I eat something". "If it is hot, I take my sweatshirt off". Prompt students for their own examples.

Conditional statements in Scratch are in the Control category because they control the flow of the program.

A conditional statement has something we do ***if it is true*** and something **else *if it is false***.

For example, **IF** the light is green **THEN**
You can go
ELSE
You must stop

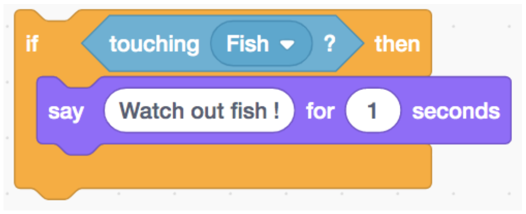
How do we know the fish touched the diver? In Scratch under the sensing category is an instruction called 'if touching … ?' . Notice there is a question mark at the end. This is the first part of a conditional statement, namely the question.



Click on the diver sprite. To help with understanding of boolean blocks:

Place the **touching Fish** condition in the script working area and test its value by clicking it (it should False

Then move the fish to the diver and click on the blue condition block again to see the value (it should be true).



Now place the block inside an if-then block. What do we want to happen if the fish touches the diver?

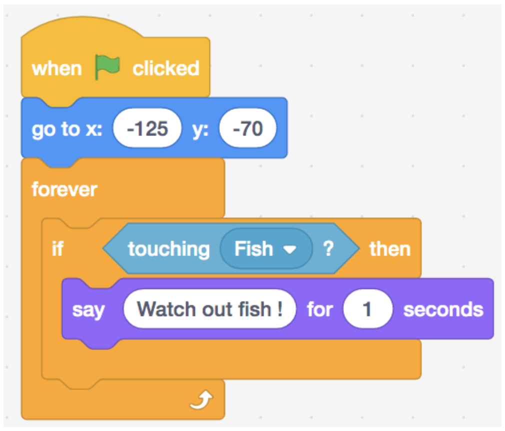
The diver says "Watch out fish!", so we add a say block inside the if statement.

Students' Turn:

Students code the if-then block. Remind them that this code is for the diver sprite, not the fish.

Continue your demonstration:

Next we need to check if the fish touches the diver all the time, not just once. So we need a forever loop around the code to check if the fish touched the diver. Lastly, we can't assume the diver is in the correct location when the program starts. It is always a good practice to add code to put the diver in the correct location. So we add a go to block at the beginning of our script.



```
when [flag] clicked
go to x: -125 y: -70
forever
    if touching Fish ? then
        say Watch out fish ! for 1 seconds
```

Run the code to test if it works.   Make any adjustments if needed.

Adjustments that might be needed:
Diver needs to jump higher or for more time if it is too difficult or less time if it is too easy not to touch the fish.

Students' Turn:

Students code the final instructions and test their program. Remind students to save their project with a name of their choosing. Review all the code with students if they are running into problems.

4. Wrap Up and Reflections

Reflection Points:

- What did you like about coding this game?
- What new things did you learn?
- What is a conditional statement?
- Why do we need a forever block around the 'if touching fish' code?