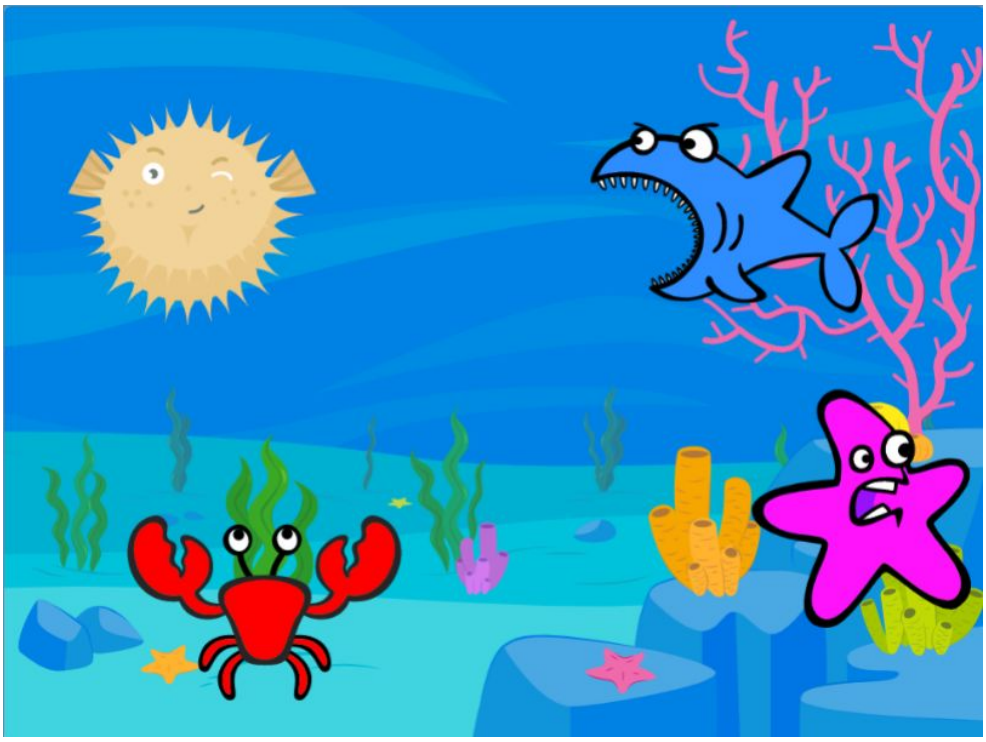
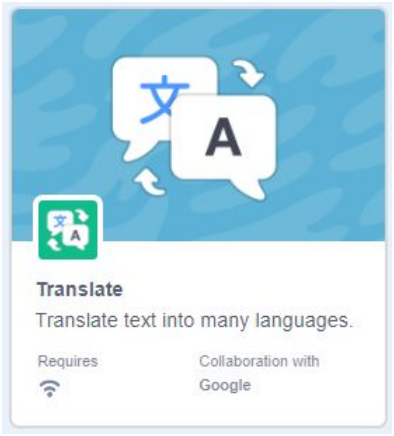
Aquatic Translators

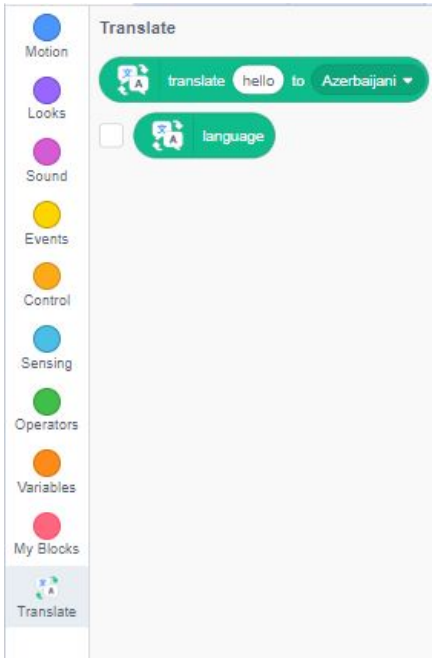


In this project we are creating a translator application by making use of the Translate extension that is available on Scratch. If you look at the screenshot above you will see 4 different sprites. Each of these sprites will be able to translate a different language. All the user needs to do is click on a sprite when they want to translate a phrase from English. Since we have 4 sprites we will be able to translate 4 different languages. The crab will be speaking Spanish, the pufferfish will be speaking German, the starfish will be speaking Swedish, and the shark will be speaking Italian. To give our application more detail, each sprite will have their own unique script for animation. Now it's time to proceed to step number 1!

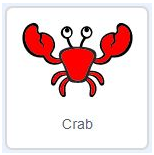
1. Our first step is to add the Translate extension to our project. Click on the add extension button in our menu, locate the translate extension, and double click it to add it to our project.



After we add the Translate extension, you will now have access to the Translate blocks.

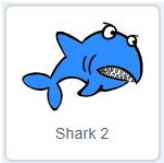


2. Now we can add our game sprites to our game. Navigate to the sprite area and add the sprites below.



Crab x1

Pufferfish x1



Shark 2 x1

Starfish x1

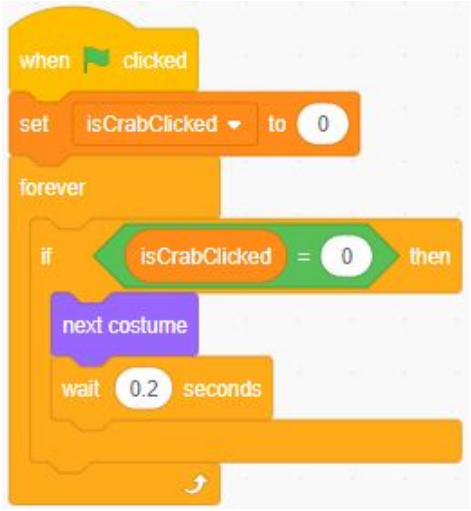
3. It is now time to create our variables. Go to the variables menu and click on “Make a variable”. We will need to create 4 variables. The names are shown below.

   isCrabClicked
   isPufferClicked
   isSharkClicked
   isStarClicked

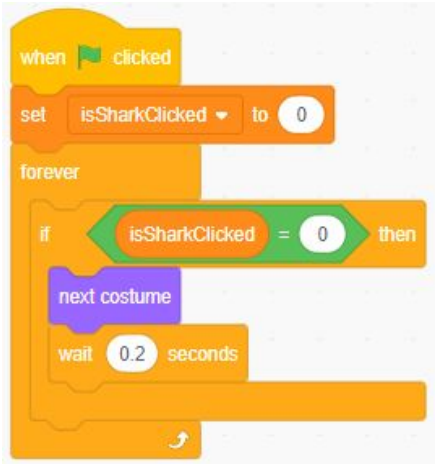
   These variables will be used to keep track of when our sprites are clicked on. We do this because we want our sprites to stop animating and start translating when they are clicked.

4. Before we make our sprites translate, we can make them come to life and animate. Each sprite will have 2 different scripts for animation. One will switch the sprite’s costume and the other will make the sprites move around the screen. We can start by adding the code that switches the sprite’s costume. Add the following code to each sprite.

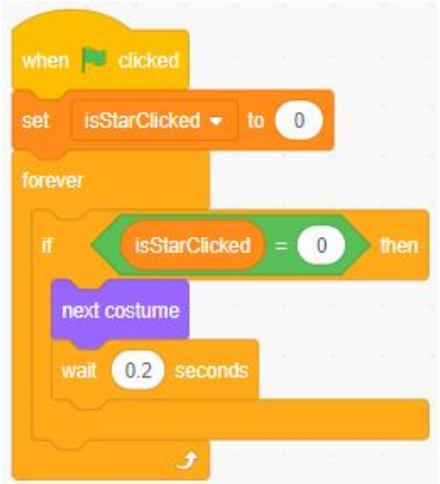
   Crab



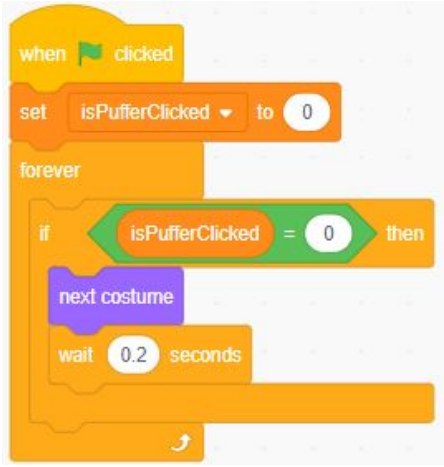
Shark



Starfish

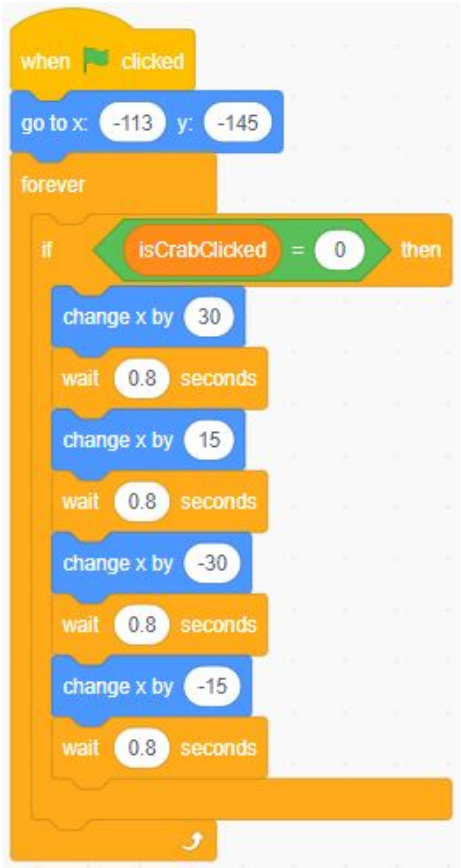


Pufferfish



Each of these scripts are very similar. The main difference is the variable that is being used for each script. The first block after the “When green flag clicked” block basically sets the variable to 0. We only want our sprites to animate when this variable is equal to 0. Setting this variable to 0 when the game starts makes our sprite start animating right away. Inside our forever loop we have an IF statement to check when our variable is equal to 0. If it does equal 0 then the sprite’s costume will begin to switch. We add a “Wait 0.2 seconds” block to slow down our costume animation.

5. In this step we are going to add the code that will make our crab move. Go to the sprite area and select the crab sprite. Add the code below to the crab sprite.

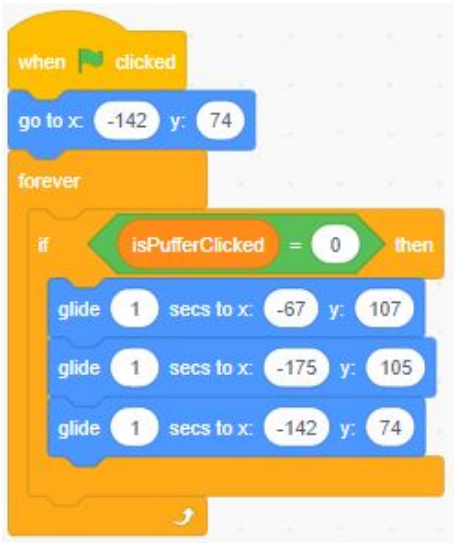


After the “When green flag clicked” block we use a “go to x:-113 y:-145” to set the position of the crab when the game begins. After that we have a forever loop with an IF statement inside to ALWAYS check the value of the isCrabClicked variable. If the variable is equal to 0 then the code inside our IF statement gets executed. All the code inside of the IF statement is doing is moving our crab left and right along the x-axis.

6. Next select the shark sprite in the sprite area. Add the code below to our shark to get the shark moving around the screen.

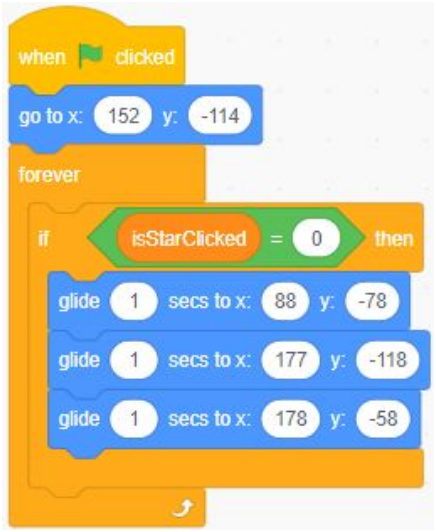
After we set the sharks position using the “go to x:86 y:66” block we have a “set rotation style left-right” block. This bock enables the sprite to face right when pointing 90 degrees and face left when pointing -90 degrees. Just like the code in our crab sprite, we have an IF statement inside of a forever loop to always check when isSharkClicked is equal to 0. If this statement is true then the code to move our shark sprite gets executed. The first block, “point in direction 90” makes our shark face right. After that we have the shark glide to the right. Then we use a “wait 0.8 seconds block” to create a small pause before he moves to the left. The “point in direction -90” block makes the shark face to the left. After that we then use a glide block again to make our shark glide to the left.

7. Now we can add the code for the pufferfish. Select the pufferfish in the sprite area and add the code below to the sprite.



Just like in the previous 2 sprites we use a “go to x:152 y:-114” block to set the position of our sprite when the game begins. Inside the forever we have an IF statement that checks if the star is not clicked on and the isStarClicked variable is equal to 0. If isStarClicked does equal 0 then we have 3 glides bock that make our starfish sprite glide to 3 different positions.

8. The code that makes our pufferfish move is very similar to the code we used for the starfish. Select the starfish from the sprite area and add the code below.



9. Now that we are done with the animation code for our sprites let's test it out to see if our sprites are animating correctly. Click the green flag to see how the sprites are animating. If it looks good then proceed to the next step.

10. How about we add a background to our game. Since we are using sea creatures as sprites let's choose a backdrop that sets our game in the ocean. Go to the stage area and click on choose a backdrop. Search either of the backdrops below and add it to your game.

11. We are now close to finishing our game. All we need to do is add the code that enables our sprites to translate words from english to another language. We can begin with the crab sprite. Select the crab sprite and add the code below.

The first block we need is the “when this sprite is clicked” block. This block can be found in the events menu and basically makes it so the rest of the code gets executed only when the crab is clicked on. Next we want to use a “set isCrabClicked to 1” block to make our sprite stop animating. We then use a “say” block from the looks menu to say what language our sprite is able to translate to. Next we will need an “ask” block from the sensing menu. Inside of our ask block type the question “What would you like me to translate?”. Then add another “ask” block from the looks menu and type “In spanish we say...” in our ask block.

The next block is actually 3 blocks combined into 1. Below is an image of the 3 different blocks we will need for this section of our code.

The “answer” block will hold the value of whatever the user types into the ask box. The say block can be found in the looks menu, and the translate block can be found in the Translate menu. For the crab we want to make sure we select Spanish from the dropdown menu in our translate block. Finally, at the end of the script we set the variable “isCrabClicked” to 0 so the sprite can start animating again.

12. This is our final step! All we need to do is add the code for the other 3 sprites. Add the code below to each sprite.

Shark

Starfish

Pufferfish

The important thing to keep in mind are the languages that need to be set for each sprite. Double check to make sure the crab is able to translate Spanish, the shark translates to Italian, the starfish needs to be set to Swedish, and the Pufferfish will be translating into German.

GREAT JOB!

Our translator application is now complete! Give it a try and see what happens when you click on a sprite. Each sprite should display a box where the user can type in a word or phrase they would like to translate. All you need to do is click on the blue check mark in the input box or press enter when you are finished typing your phrase. Then our sprite will respond by saying our translation in the chosen language.

Optional

Try making your game loop music. Navigate to the backdrop and add code to our backdrop that makes our game play music. First you need to choose a sound. In the sounds menu locate the “loops” tab. In there search for a loop that you want to add to your game.

The code below is added to the backdrop icon. This makes music loop while our game is running.