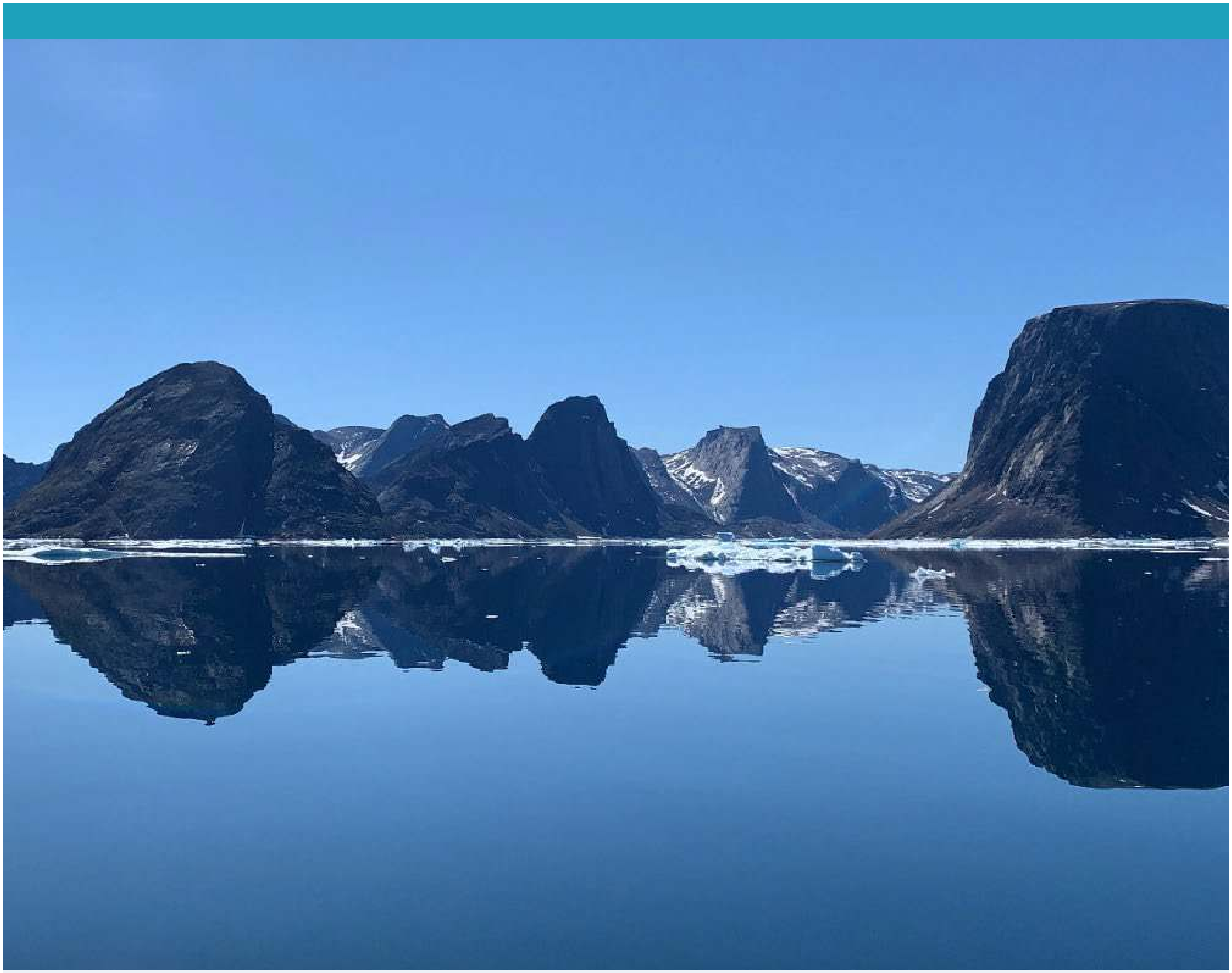


Literacy Through Scratch Story Games

TEACHER'S GUIDE

Heritage Canada

Prepared by Ilitaqsiniq (Nunavut Literacy Council) and Pinnguaq Association

CONTENTS

In a collaboration between Ilitaqsiniq - Nunavut Literacy Council, and Pinnguaq Association and informed through community consultation, a learning resource has been created using the free, open-source digital game design tool Scratch. This resource will allow Inuktitut language educators and their students to create their own small-scale literacy games to engage their learners and foster learning in their classrooms.

GAME STUDIO

Access Scratch Game Studio at **scratch.mit.edu/studios/29216436**. With great appreciation to Leo Karetak for his beautiful artwork for the game sprites, and to Jonah Oolayou for his stunning photographs featured in this guide and as assets for the Scratch Literacy Games.

TEACHING AND LEARNING INUKTITUT THROUGH GAMES

Prepared by Ilitaqsiniq in partnership with Pinnguaq

The Scratch games developed with Pinnguaq take teaching and learning Inuktitut to the next level. Educators will now have access and flexibility to adapt and integrate many parts of language development into these games, and students will get to learn interactively while playing. Because the Scratch platform does not require the internet it allows equal access to educators and students in remote communities. This will help Inuktitut strive in today's technological world.

LANGUAGE TEACHING THROUGH INTERACTIVE STORY GAMES

Teachers' Guide • Scratch Literacy Project • March 2021
Prepared by Pinnguaq in partnership with Ilitaqsiniq

"Play has an important role in the physical, social, emotional, language and cognitive development of children and in essence, it is a learning experience. Play contributes to children's general personality development, allowing them the opportunity to practice their linguistic, cognitive, and social skills. Play is also associated with creativity, especially the ability to be less literal and more flexible in one's thinking." (**Theories Surrounding Learning Through Play**, 2015).

In play, learners are fully absorbed in the learning activity, focussed and engaged. Play creates an enjoyable learning environment. Learners are motivated to continue playing and learning and eager to return. The Scratch platform is accessible for both educators and students to create their own interactive language story games, offering the opportunity to develop creativity and coding skills alongside language.

Scratch story games are interactive. Story games that involve characters talking with each other and to the player; matching words, pictures, and sounds, and discovering knowledge about the land and language can be created and shared within and amongst communities.

Some of the benefits of playing games to develop language literacy (**FluentU**):

- Fun and enjoyable
- Provides meaningful context for language use
- Increases learning motivation
- Promotes communicative competence
- Reduces anxiety of making errors
- Integrates numerous linguistic skills: reading, listening, speaking, and writing

SCRATCH AS A LANGUAGE TOOL

Introduction to Scratch

Scratch was developed as a tool to teach people how to code using blocks that snap together very much like Lego blocks can be put together to create something.

> Scratch is both an accessible tool for creating and offers a platform for sharing multilingual, interactive stories and games. Scratch offers the opportunity to create and share Inuktitut language resources in an engaging way.

"Your students can use Scratch to code their own interactive stories, animations, and games. In the process, they learn to think creatively, reason systematically, and work collaboratively—essential skills for everyone in today's society. Educators are integrating Scratch across many different subject areas and age groups." (**Scratch for Educators**)

We can use any language and any writing system, such as syllabics in Scratch. Audio is built into the platform and you can upload or record your own sounds and speech right in the story game. Students can read, listen, speak and write within Scratch story games. Scratch story games can be shared and played online and offline which opens up the opportunity of creating a shared bank of interactive story game Inuktitut language teaching tools.

SCRATCH BASICS

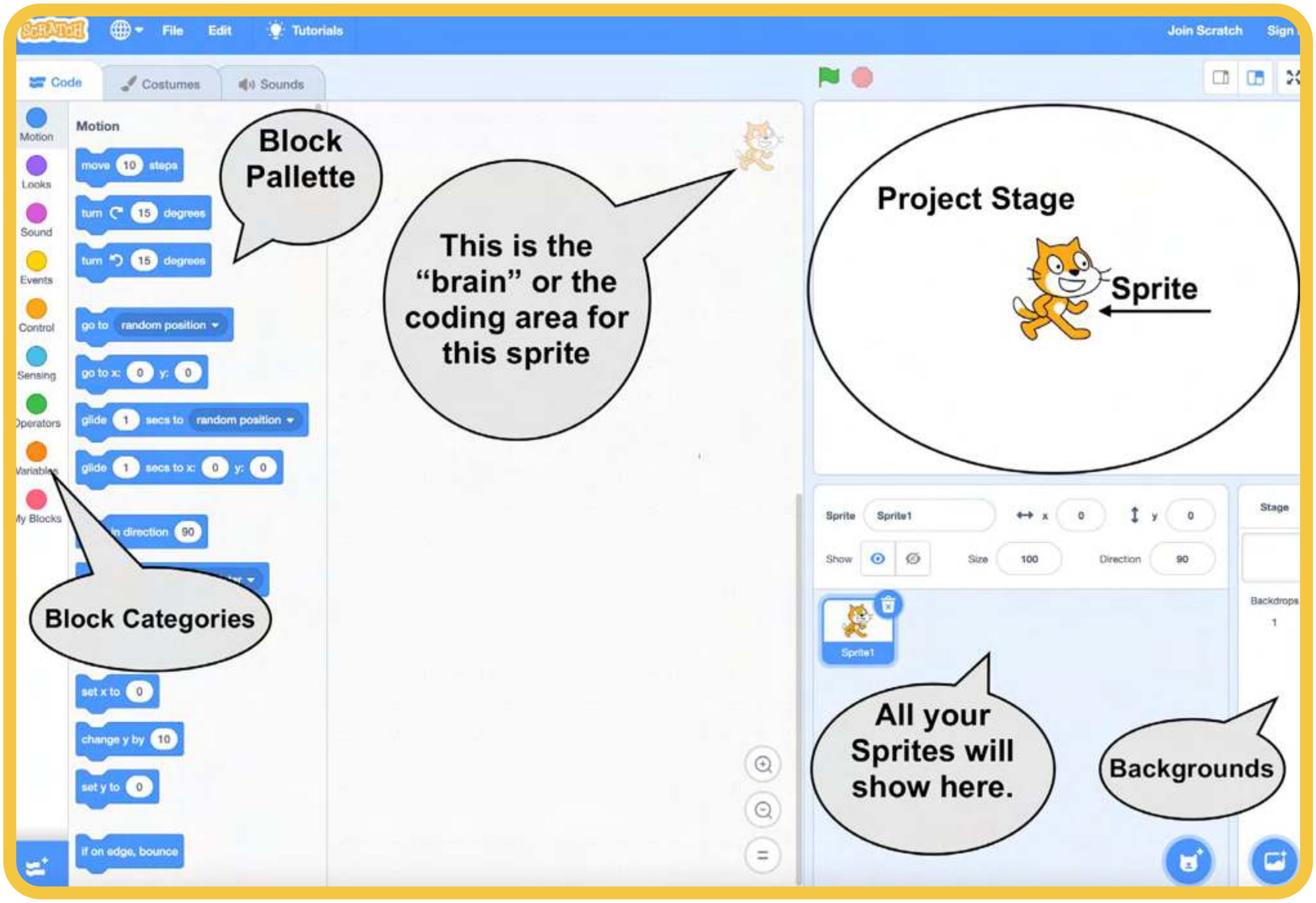
The following step-by-step guide will help you to learn about the Scratch working environment.

Welcome to Scratch

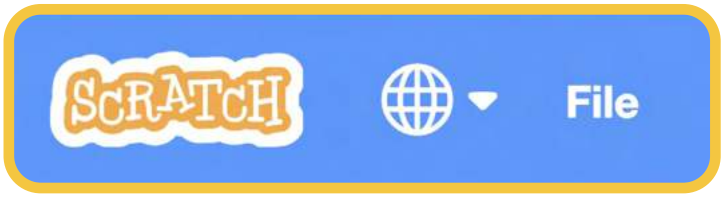
1. **Open the offline Scratch 3 app or the online Scratch editor at scratch.mit.edu (choose Start Creating)**
   - **a)** If you would like to save your Scratch projects online, you can create an account.
   - **b)** If you will be working offline, you can save your projects to your computer.
   - **c)** These Scratch (sb3) files are shareable by email, can be loaded onto a USB or saved to a computer.

2. Five sections of the screen

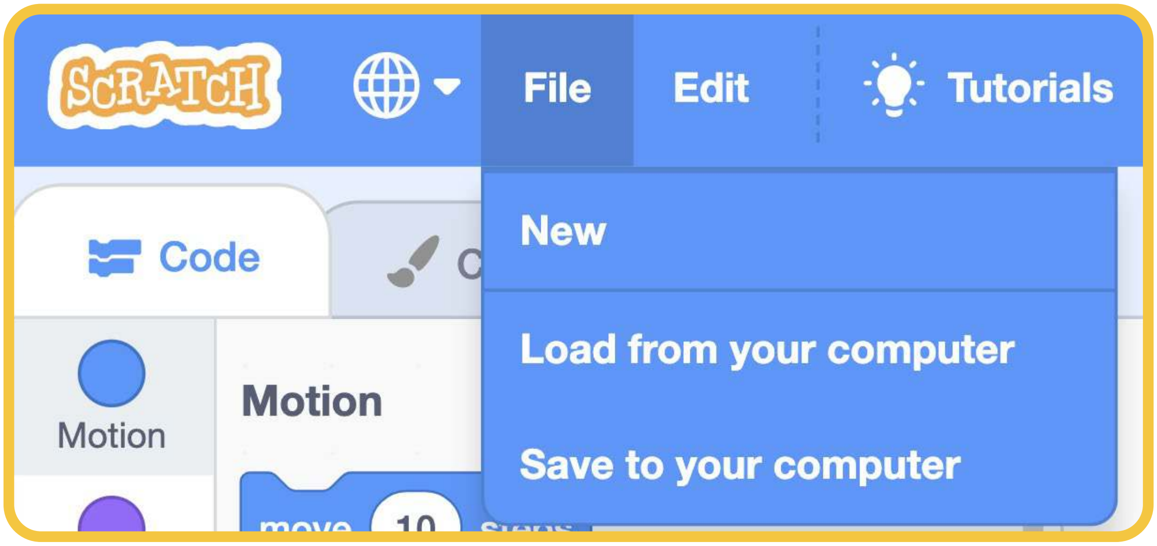
Here you are introduced to the five key sections of the working environment in Scratch.



The File Menu



We will start by introducing the basics. We want everyone to start in a new Scratch Project. By default, you will have, but to be sure, let's look at the "File Menu". There are a few options here.

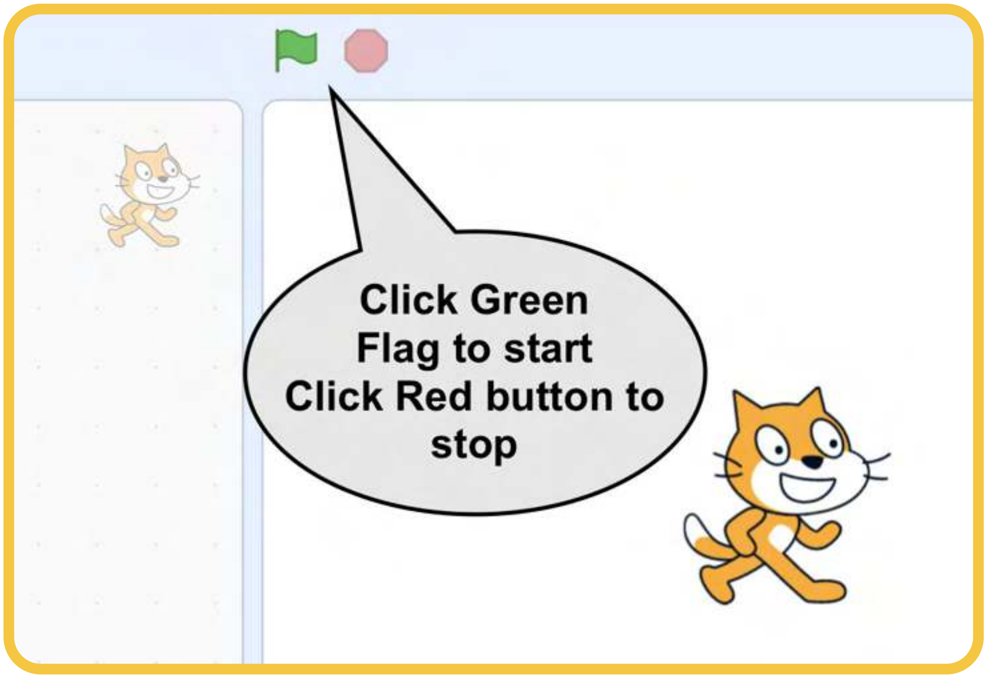


🌐 ▾	Change your language *Note: Indigenous languages are currently not available*
File - New - Load - Save	*; You will use these there commands the most* - Create a new project from Scratch - Load an existing project (sb2 and sb3 files) - Save your project
Edit - Restore - Turbo - Mode	*; You will not use these commands* - Restores project to last save point - Makes your project move at 50 times the speed
💡 Tutorials	Open up a menu to do a series of tutorials to teach yourself Scratch

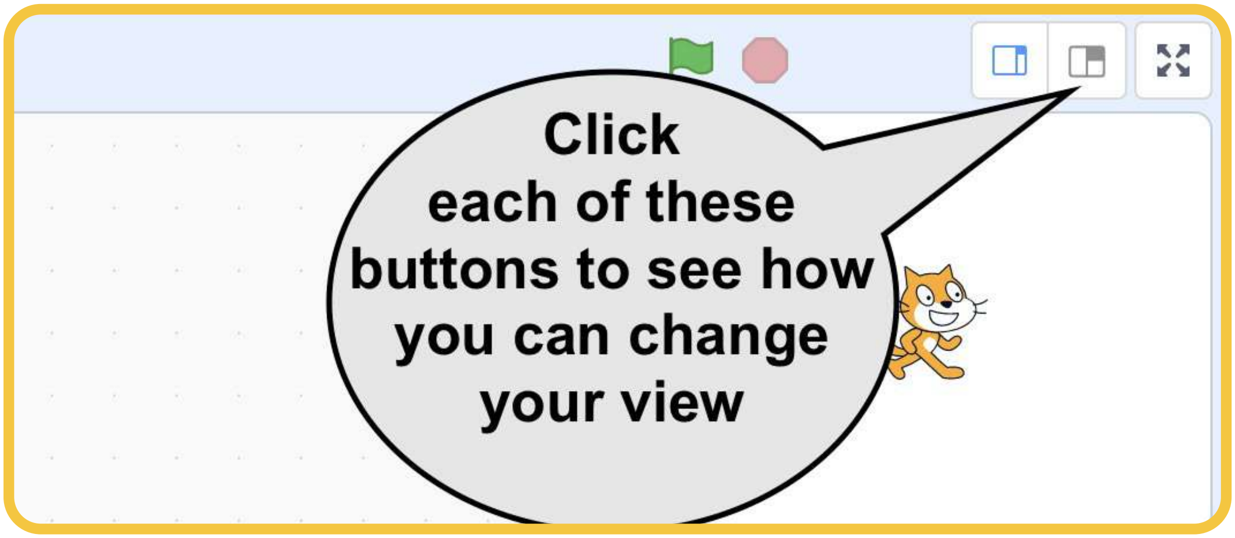
The Project Stage

This is where the actual project runs. You will only see the cat there for now. As we create a program, this is where it will run so we can see the changes we make, in action.

Click the Green Flag This is what will start your program. Right now if we press it, it does nothing. That is because you haven't yet programmed it to do anything.



You can also change the size of your working area here (2) and make the Project fullscreen

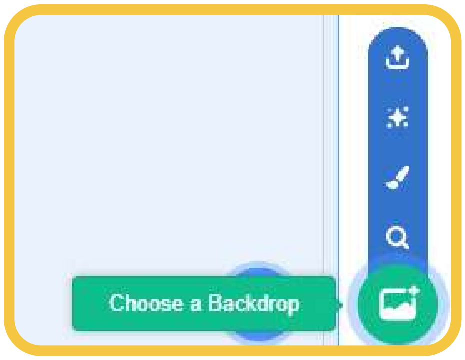

Click each of these buttons to see how you can change your view

The Sprite/Backgrounds

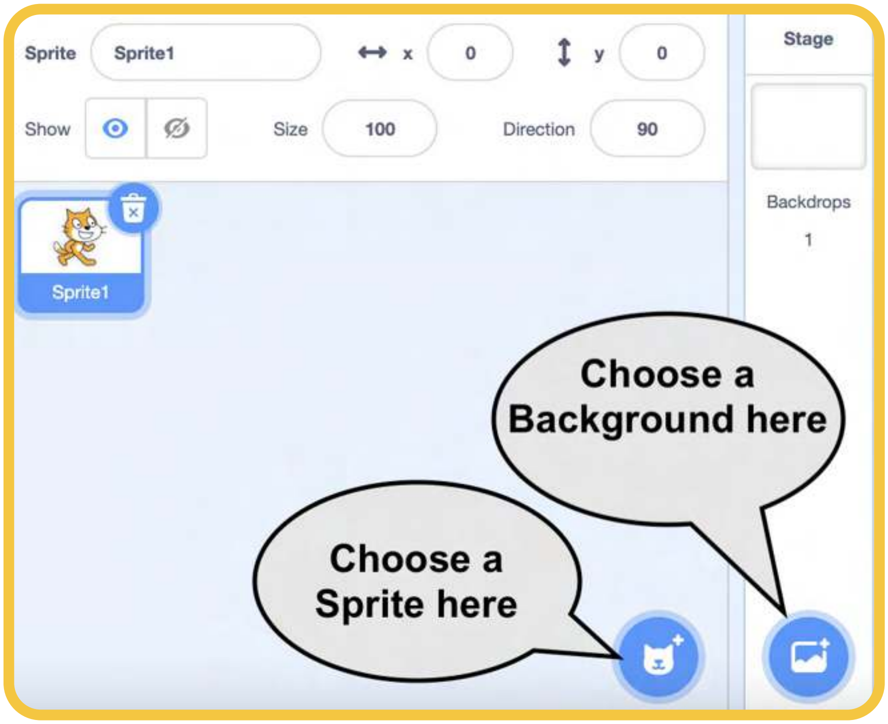
Core Question: *What is the difference between backgrounds and sprites?*

We have two things here;

- **Backgrounds:** As the name suggests these are the backdrops, the visual settings that make up the project. Backdrops cannot move. You can select a background from the Scratch library by clicking "Choose a Backdrop"



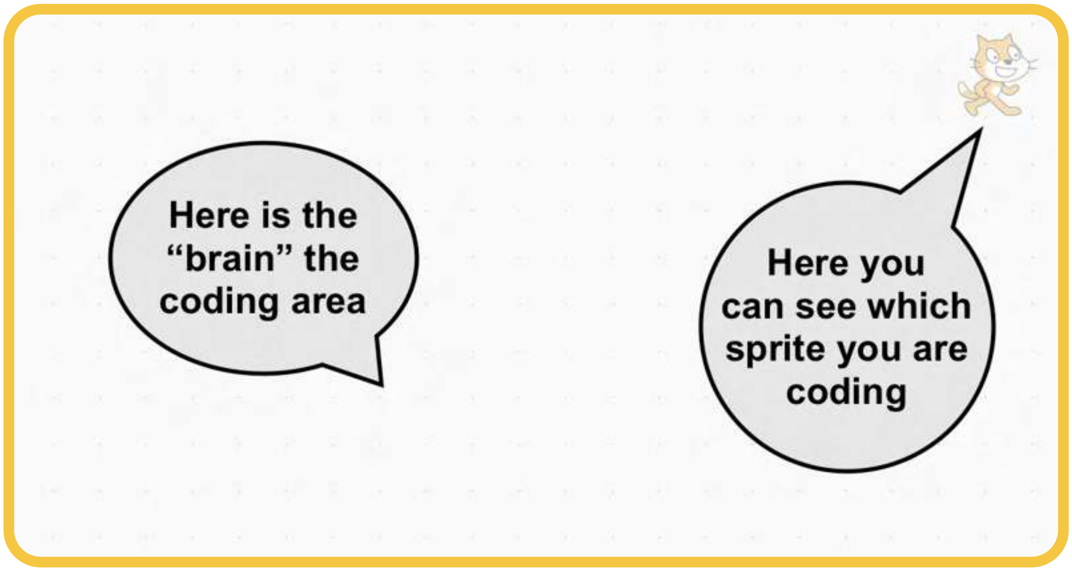
- **Sprites:** These are like the characters of a project. Sprites are anything that moves, is controlled, or is animated.



Right now you have 1 sprite. He is a cat. You can keep him.

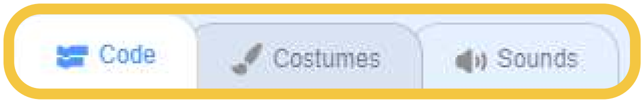
The “Brain,” the Coding Area

The big centre of the project is what we call “The Brain” of the project. Right now you are looking at the Cat’s brain. It is empty. You can fill it with any commands and it will do exactly what you say. Computers always do just exactly as you tell them. Coding is the language you use to communicate with a computer.


Here is the “brain” the coding area
Here you can see which sprite you are coding

Code/Costumes/Sound

In this introduction, we will focus on “Code”, which should be highlighted for you by default.

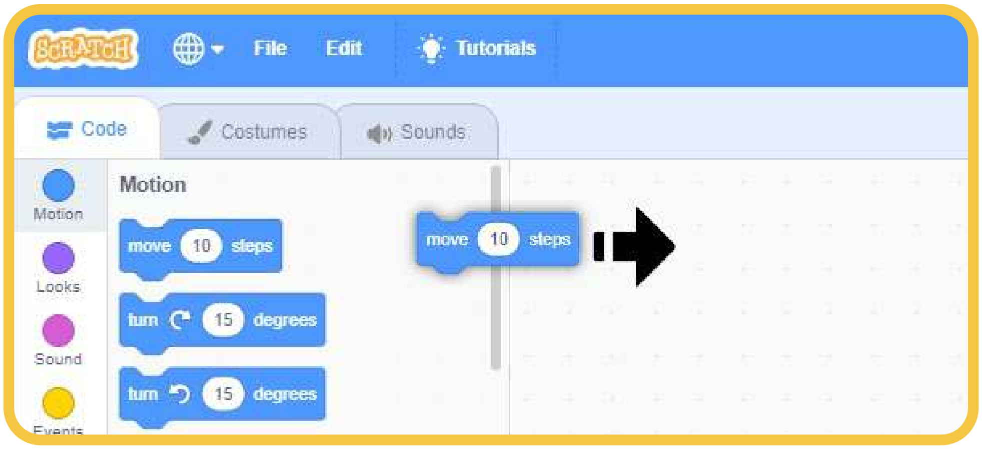


This is where we can grab things to add to the cat’s brain! We can grab commands and the cat will do what we tell him to.

Make the Cat Sprite Move

Programming in Scratch works with just dragging and dropping commands into the cat’s brain. You can join them together like lego blocks to form “algorithms” or longer series of commands that accomplish one task.

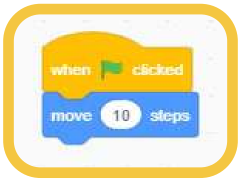
Drag “Move 10” steps from ● Motion into the cat's brain.



If you click on the cat, or the command “Move 10 Steps” in his brain you will see the moves 10 “steps”. (Steps defined by Scratch using “pixels”).

Now you can make it a real program by adding “When Green Flag Clicked”

From ● Events let's grab when 🏁 clicked and ‘click' it into the top of “Move 10 Steps” in the Cat's brain.



```
when 🏁 clicked
move 10 steps
```

Now when the user clicks on the “Green Flag” (as introduced before) the cat will move 10 steps.

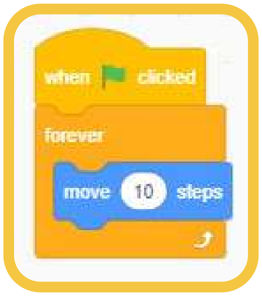
Celebrate! You have created your first program! It's a simple program, but you've done it! Pat yourself on the back!

Forever Loops

Let's make this program better.

What do you think a “Loop” might be? It's something that goes around, something that repeats, something thats ending is also its beginning.

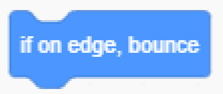
From Control let's grab a "Forever" command and place it around "Move 10 Steps".



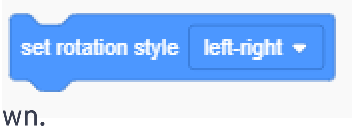
```
when [flag] clicked
forever
    move (10) steps
```

Now when you hit the green flag, your cat walks forever. But remember, computers are dumb. He will literally walk forever and off the screen.

The user can grab the cat by the tail and bring him back to the centre of the screen. Now you can add additional commands;

- From Motion add  if on edge, bounce and place it ***in the forever loop.*** Whenever the cat detects an edge, he will turn around and start walking in the other direction

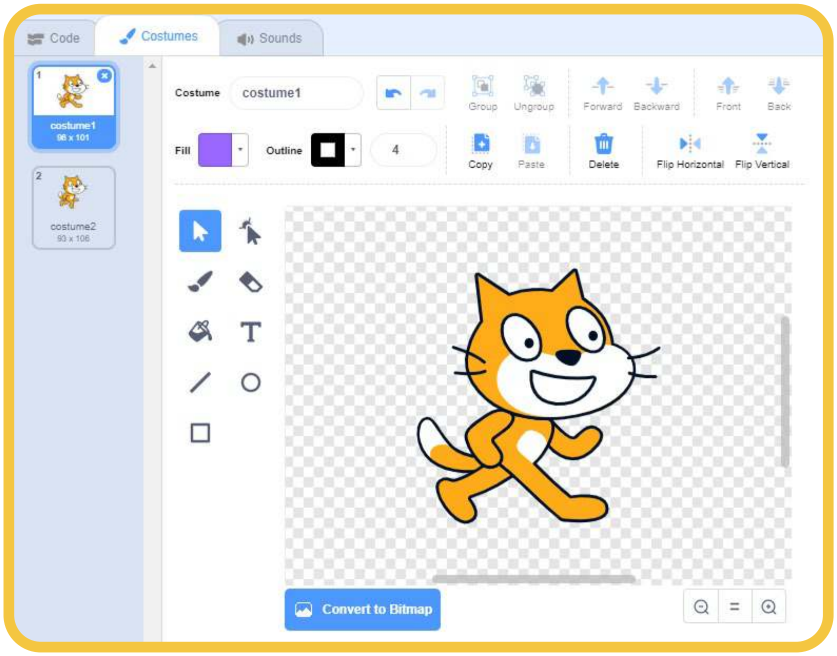
  But now he flips upside down.

- Again from Motion add  set rotation style left-right ▾ and place it ***in the forever loop.*** This will stop him from flipping upside down.

Animation

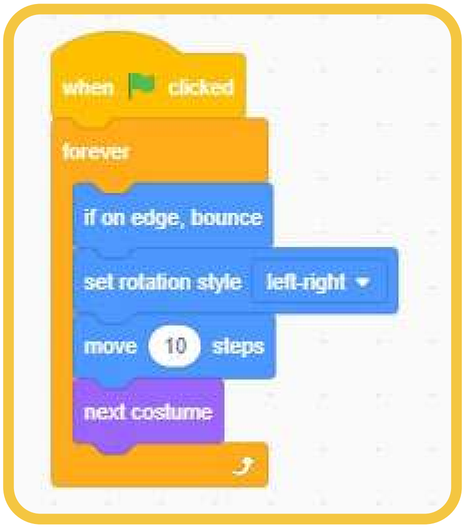
Let's make it look like the cat is walking. In Scratch 'frames of animation' are called "Costumes". The sprite puts on a "different costume" to simulate movement and if we flip between them fast enough the illusion holds. This is frame-by-frame animation.

If you click on "Costumes" we can see the cat has 2 costumes called costume1 and costume2



All we need to do to flip between costumes is

- Return to "Code"
- In ● Looks choose "Next Costume" and add it to the forever loop.

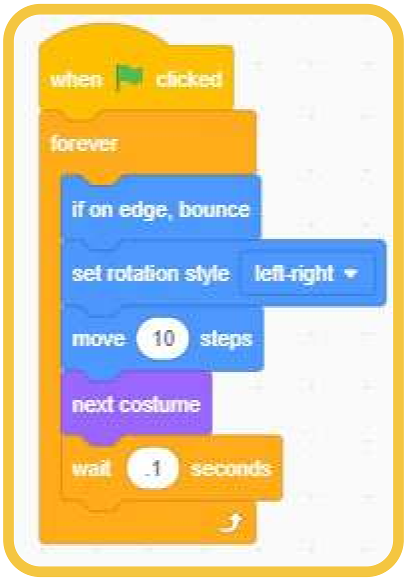


```
when 🏴 clicked
forever
    if on edge, bounce
    set rotation style left-right ▾
    move 10 steps
    next costume
```

The cat now flips *very* quickly between its two costumes and simulates walking!

To make this animation a little more realistic, let's add a "Wait" into the loop. This will cause the computer to 'take a beat' between loops.

- In ● Control choose “Wait 1 Seconds” and add it to the forever loop. 1 second is a little long, so let’s change it to 0.1 seconds and we should have a nice smooth, cat walk. Or you can play with the number to get the speed of movement you like. I like 0.2 to give the Cat Sprite a funky step.



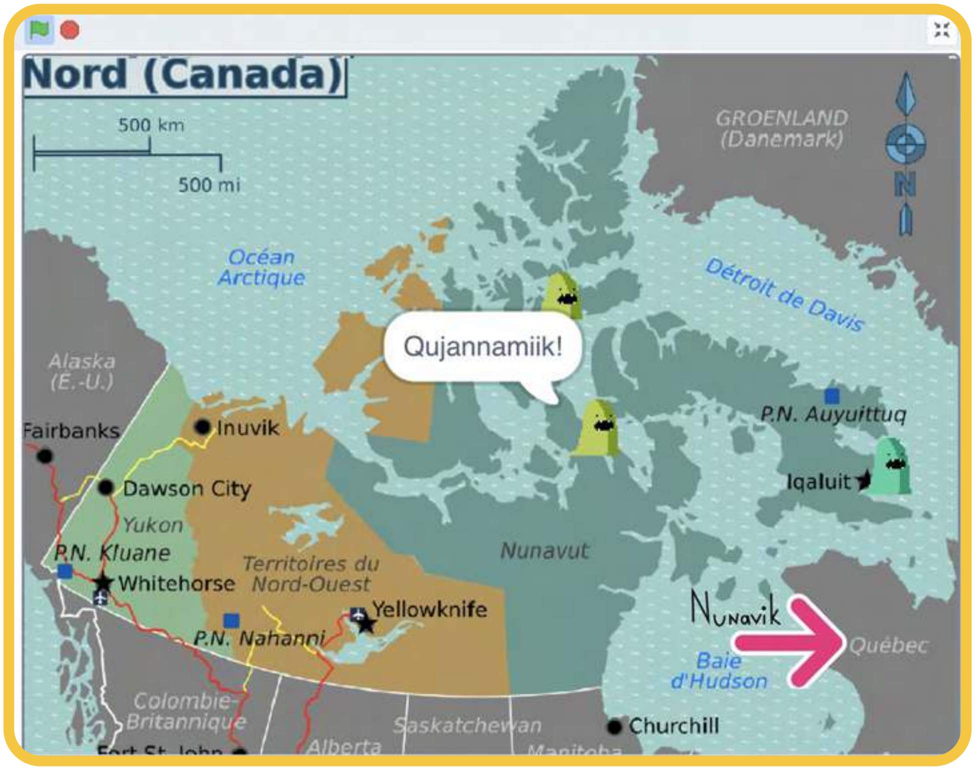
```
when 🏁 clicked
forever
    if on edge, bounce
    set rotation style left-right ▾
    move 10 steps
    next costume
    wait .1 seconds
```

To think about and try

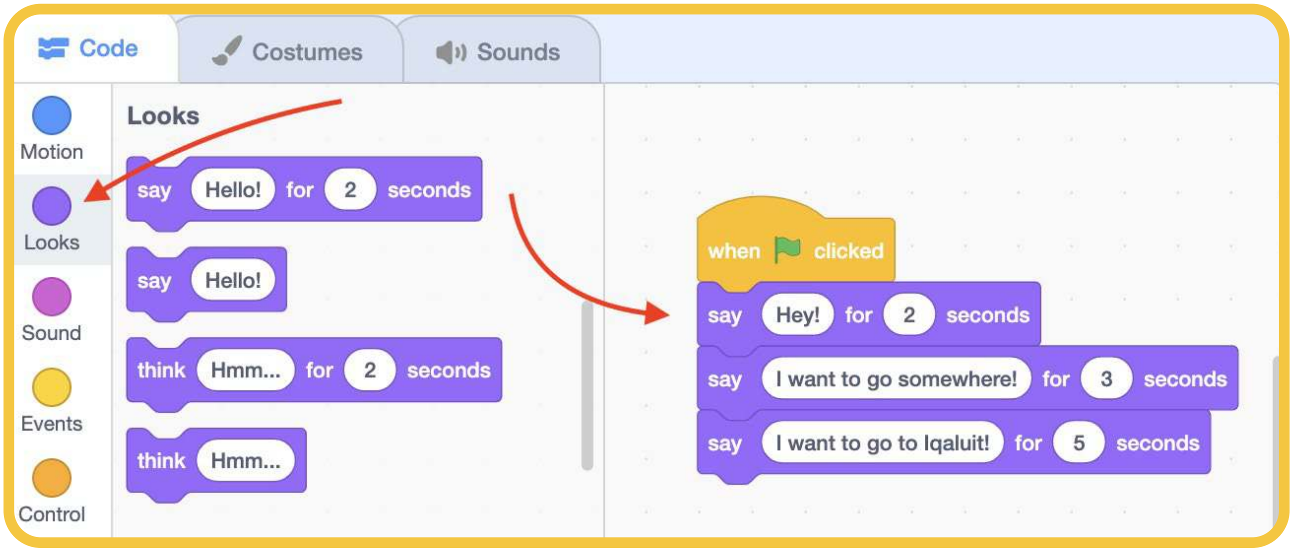
- What happens if you put new code after the Forever Loop? (Nothing: The loop never ends and as a result, that code is never reached)
- Notice everything in white are numbers you can change. What happens if you change Move 10 Steps to Move 1 step? What about Move 50 steps? What about move 100000 steps?

ADDING LANGUAGE LITERACY

Scratch as a coding tool has been translated into many languages, but unfortunately, these do not as of yet include Inuktitut. However, although we can’t code in Inuktitut in Scratch, we can have the “reader” or “player” of the interactive stories and games interact in Inuktitut. We can use the on-screen speech bubbles to have instructions and characters speak in Inuktitut and syllabics. As well, as Scratch has a built-in voice recorder feature, teachers and learners can record themselves. Sounds can be pre-recorded and uploaded to be played within the story game. Recordings of Elders and Knowledge keepers can be played within the story. All of these features make Scratch a powerful, accessible tool for creating and sharing interactive story games to help students learn and understand languages.



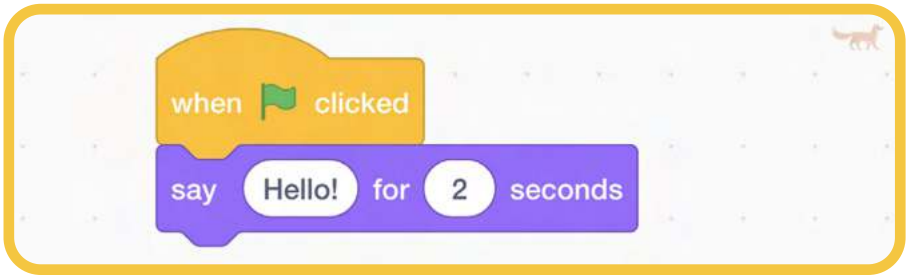
- In the Looks category, there are speech bubble blocks that can be added to the coding area. Any language and writing system can be used to type into these speech bubbles.



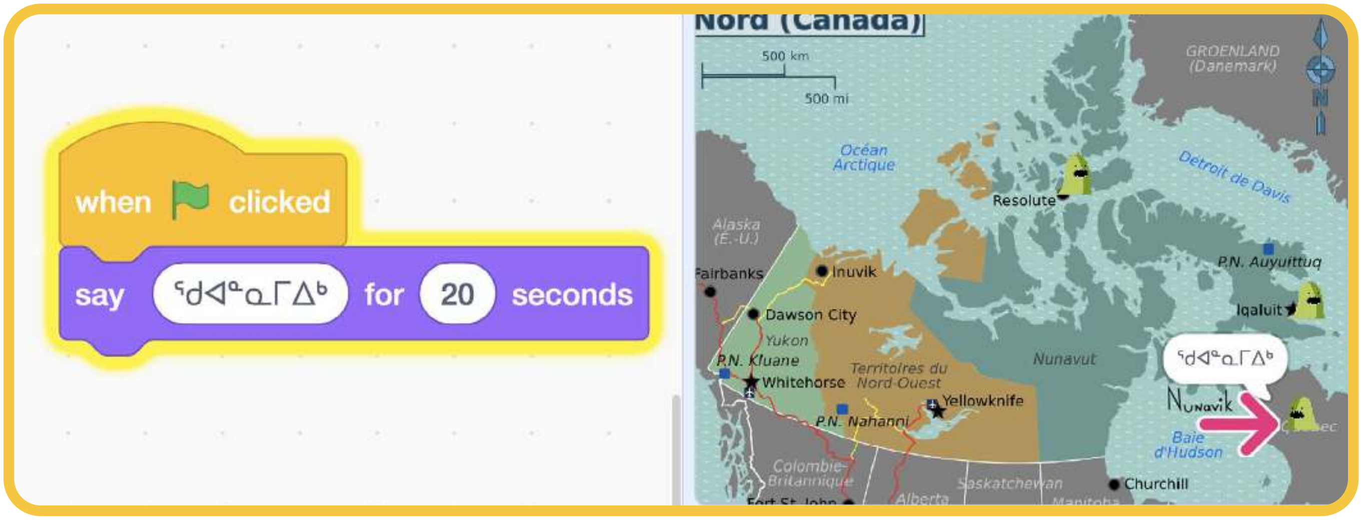
Looks

```
say Hello! for 2 seconds
say Hello!
think Hmm... for 2 seconds
think Hmm...
```

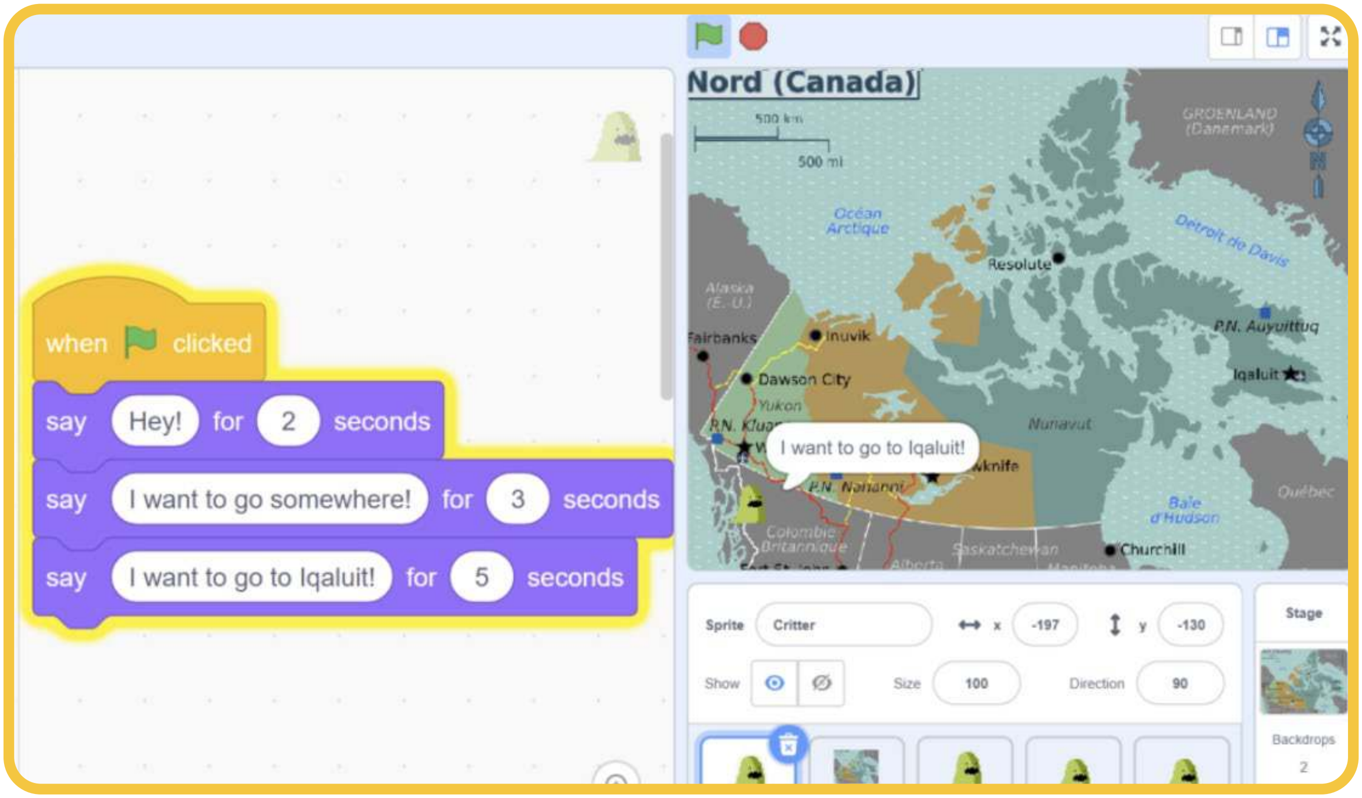
```
when ⚑ clicked
say Hey! for 2 seconds
say I want to go somewhere! for 3 seconds
say I want to go to Iqaluit! for 5 seconds
```



```
when [flag] clicked
say (Hello!) for (2) seconds
```



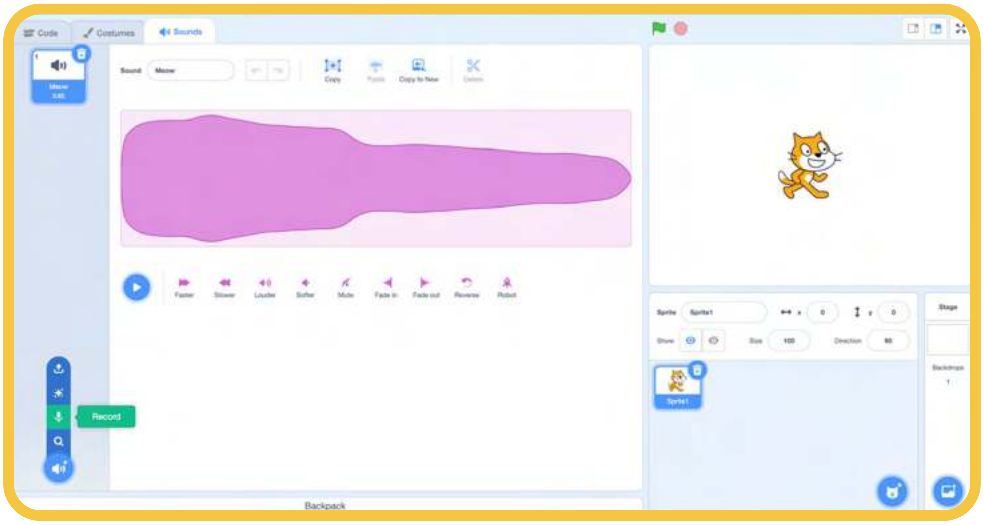
```
when [flag] clicked
say (ᖁᐊᓐᓇᒥᐅᒃ) for (20) seconds
```



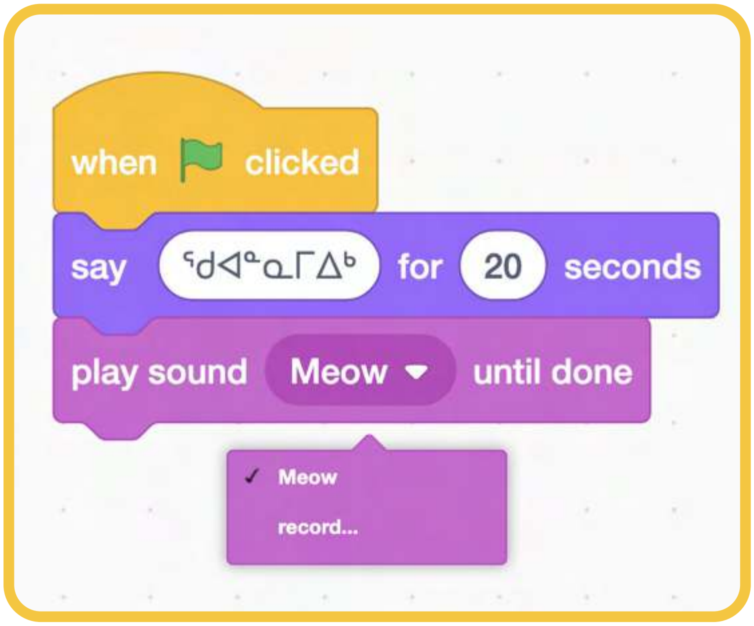
```
when [flag] clicked
say (Hey!) for (2) seconds
say (I want to go somewhere!) for (3) seconds
say (I want to go to Iqaluit!) for (5) seconds
```

Sprite: Critter x: -197 y: -130
Show Size: 100 Direction: 90

- As well, there are simple audio editing tools within the Scratch working environment which can be found on the Sounds tab. Here you can record sound, upload sound, and edit sound. The project resources include a library of recordings of syllabic sounds and some words.



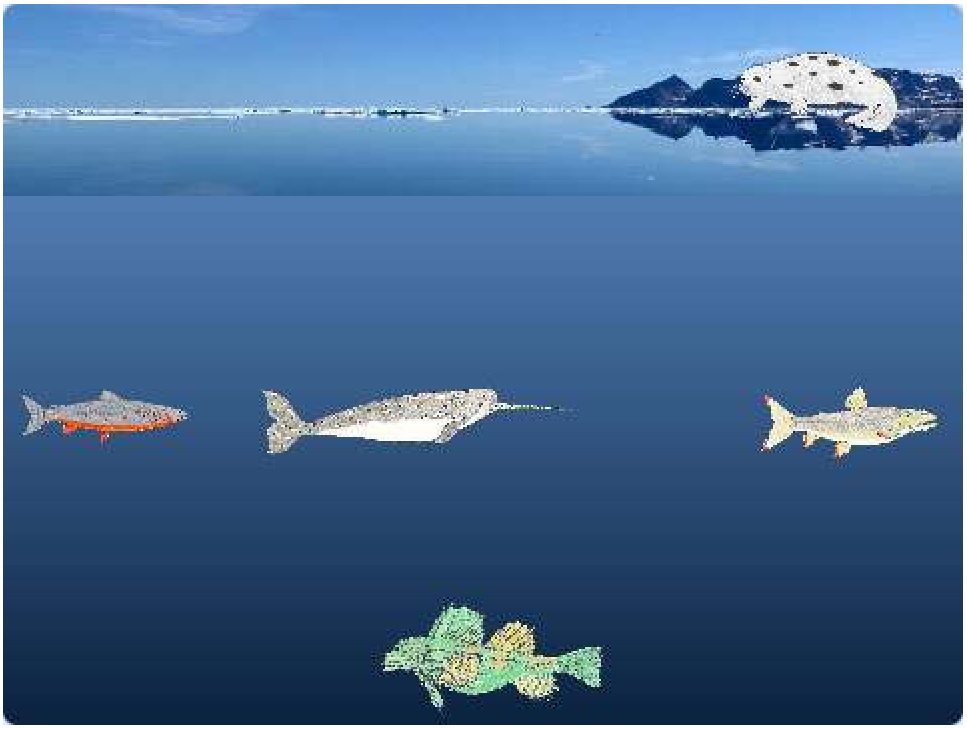
- Any sound that has been recorded will show as a choice in the code blocks, as well as the option to record a new sound.



```
when 🏁 clicked
say ᖁᐊᓐᓇᒥᐃᒃ for 20 seconds
play sound Meow ▾ until done
```

- It is easy to include both visual and auditory language in Scratch story games.

NARWHAL NAME GAME



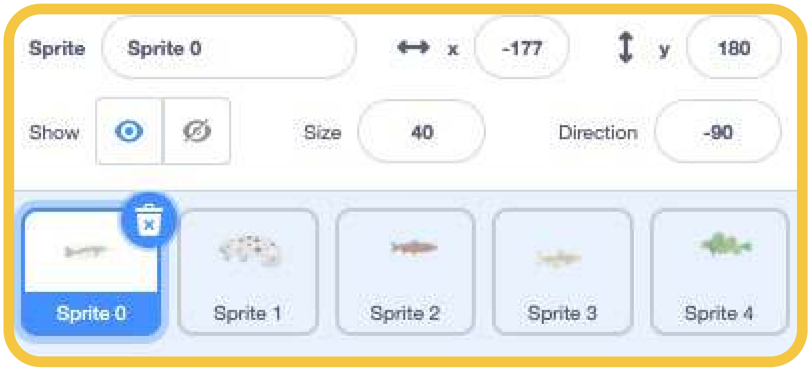
Animal Name Announcer Game in the Game Studio (**scratch.mit.edu/projects/534237341/**)

Introduction

This project is designed to help students practice their pronunciation. In this example, animals are used as the characters in the game. Students can record the pronunciation of each animal's name. The narwhal character will follow the mouse pointer once the green flag is clicked. When the narwhal touches another character, the audio recording for that character will play. The animals and background can easily be replaced with other animals, plants, objects, and scenery to work on a different topic or new vocabulary.

How the Code Works

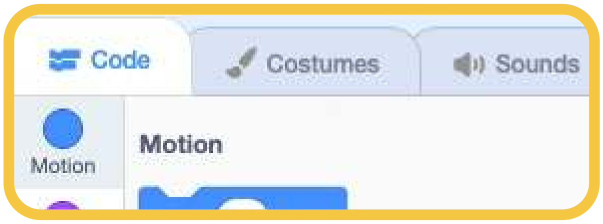
In Scratch, each 'sprite' in the game can have its own set of instructions in code. In this project, we will just be working with the code on our Narwhal character 'Sprite 0'. To view the code for Sprite 0 at any time, we can click on 'Sprite 0' in the bottom right corner of the screen.



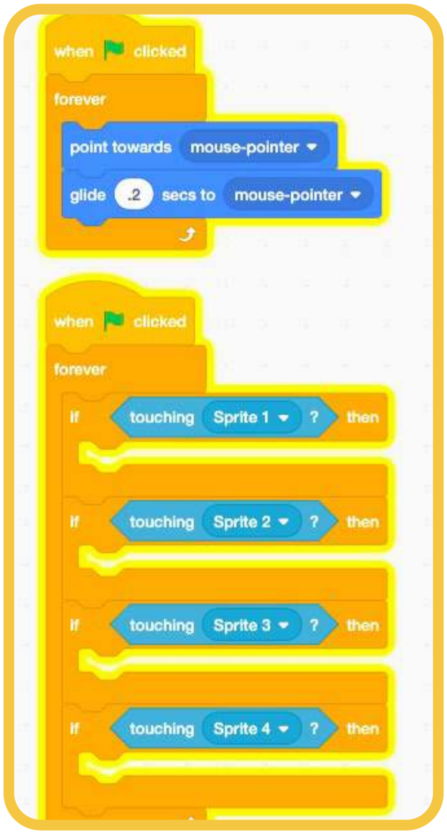
Sprite: Sprite 0 ↔ x: -177 ↕ y: 180

Show | Size: 40 | Direction: -90

Then, in the top left corner of the screen, we can click on 'Code'.

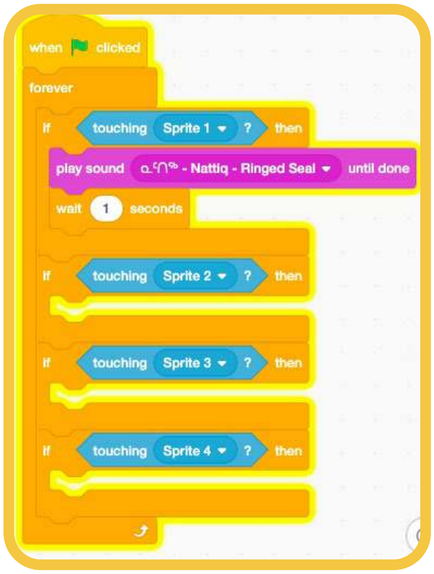


Now the blocks of code that give our Narwhal its instructions should be visible. The first set of instructions in this project tell our narwhal how to move. These instructions will begin 'when the green flag clicked' and continue 'forever' while the game is running. The middle part of these instructions tells the Narwhal to point towards the mouse pointer, glide for 0.2 seconds towards the mouse pointer.



```
when 🏁 clicked
forever
    point towards mouse-pointer ▾
    glide .2 secs to mouse-pointer ▾

when 🏁 clicked
forever
    if touching Sprite 1 ▾ ? then

    if touching Sprite 2 ▾ ? then

    if touching Sprite 3 ▾ ? then

    if touching Sprite 4 ▾ ? then

```

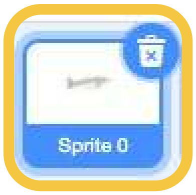
The next part of the Narwhal's code also begins happening over and over forever when the green flag is clicked. The code checks to see if our Narwhal (Sprite 0) is touching any of the other sprites. If the Narwhal is touching Sprite 1, then the recorded sound will be played to pronounce 'ᓇᑦᑎᖅ- Nattiq'. The 'wait 1 seconds' block adds a delay before the sound is repeated.



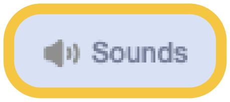
```
when 🏁 clicked
forever
    if touching Sprite 1 ▾ ? then
        play sound ᓇᑦᑎᖅ - Nattiq - Ringed Seal ▾ until done
        wait 1 seconds
    if touching Sprite 2 ▾ ? then

    if touching Sprite 3 ▾ ? then

    if touching Sprite 4 ▾ ? then

```

Adding Your Own Recorded Sounds

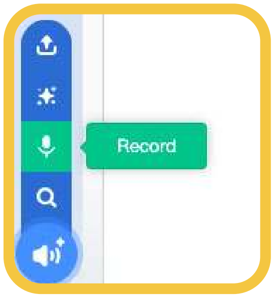
To add a sound in this project, first click on 'Sprite 0' in the bottom right corner of the screen, the Narwhal sprite in our example.



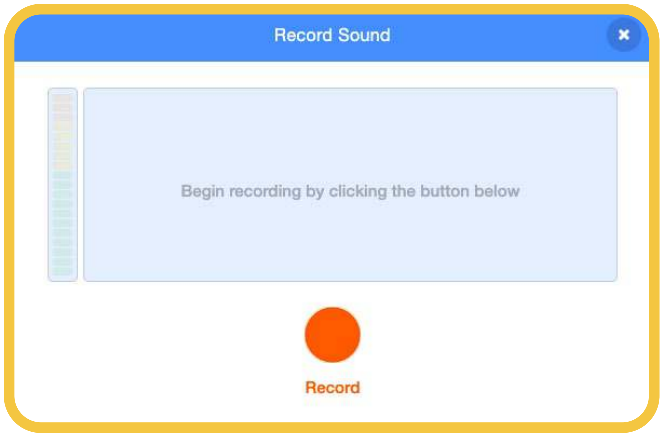
Next, click the 'sounds' tab in the top left corner.



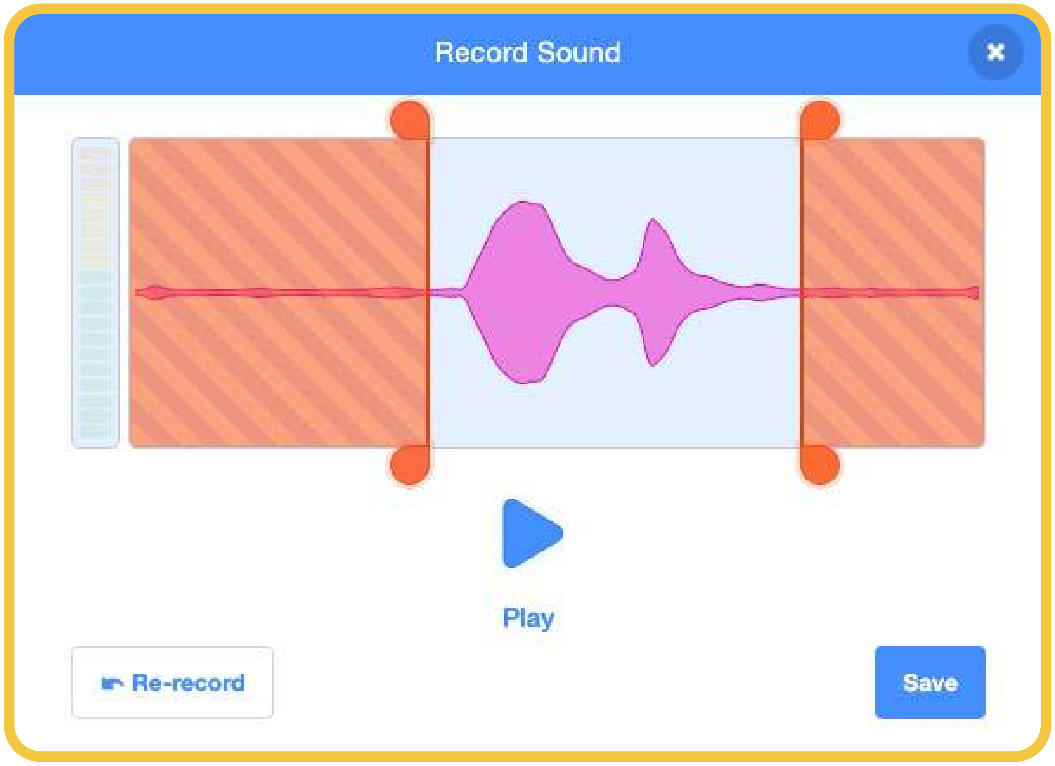
To record a new sound, hover your mouse pointer over the 'speaker' icon in the bottom left corner then click the microphone to record.



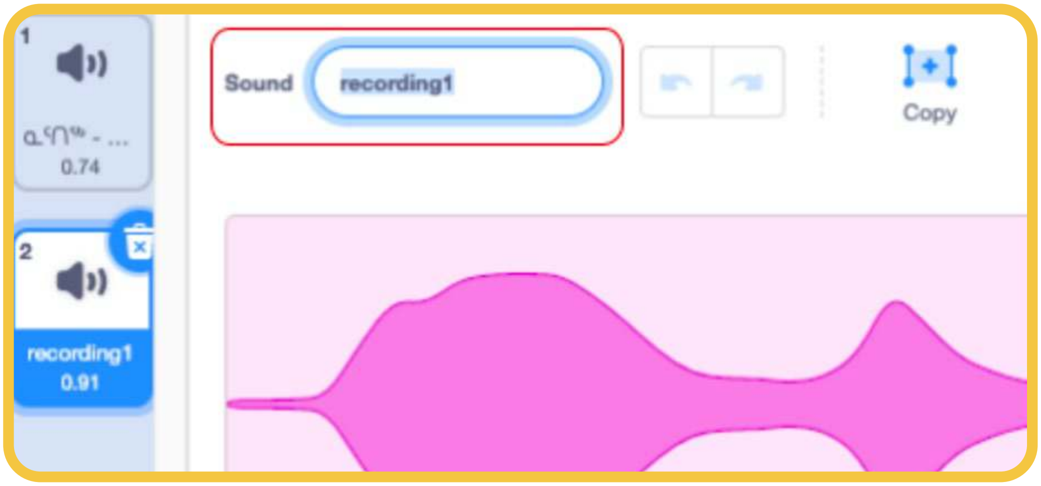
The next window to pop up will allow sound to be recorded after the red 'record' button is pressed. You may need to allow Scratch permission to access your device's microphone.



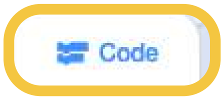
If you wish to trim the start or beginning of the recording, adjust the red sliders. This can be used to remove noise from before or after you pronounce the word for your project.



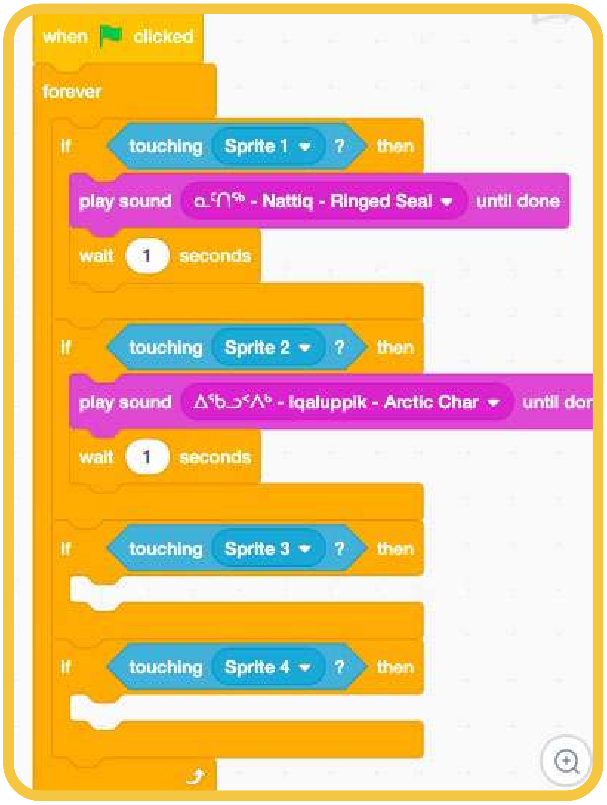
After clicking ‘save’ it is helpful to rename the recording you just made to match the word you pronounced. Click on recording1 (show in the red rectangle below) and use the keyboard to type in a new name.


Sound recording1

Click back on to the 'code' tab at the top left of the screen.

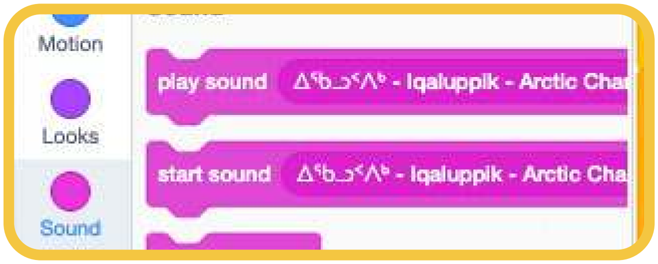


Now let's add some code so that our new recording will play when our narwhal touches the appropriate animal. In my game, I'm adding code to the instructions for the narwhal so that it will play our new recording if it is touching Sprite 2.



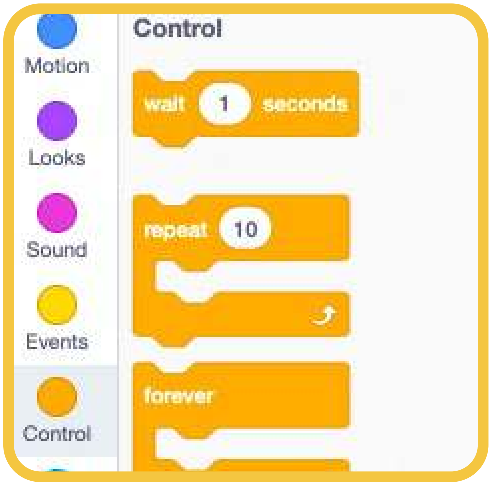
```
when 🏁 clicked
forever
  if touching Sprite 1 ? then
    play sound ᓇᕐᑎᖅ - Nattiq - Ringed Seal until done
    wait 1 seconds
  if touching Sprite 2 ? then
    play sound ᐃᖅᑲᓗᑉᐱᒃ - Iqaluppik - Arctic Char until dor
    wait 1 seconds
  if touching Sprite 3 ? then
  if touching Sprite 4 ? then
```

Click on the purple 'sound' section to find the 'play sound' block.



```
play sound ᐃᖅᑲᓗᑉᐱᒃ - Iqaluppik - Arctic Char
start sound ᐃᖅᑲᓗᑉᐱᒃ - Iqaluppik - Arctic Cha
```

Click on the orange 'control' section to find the 'wait 1 seconds' block.



This process can be repeated to add sounds for each character in the game. By adding new costumes and new backdrops, this project can be modified to help students learn new vocabulary.

MATCHING GAME

Introduction

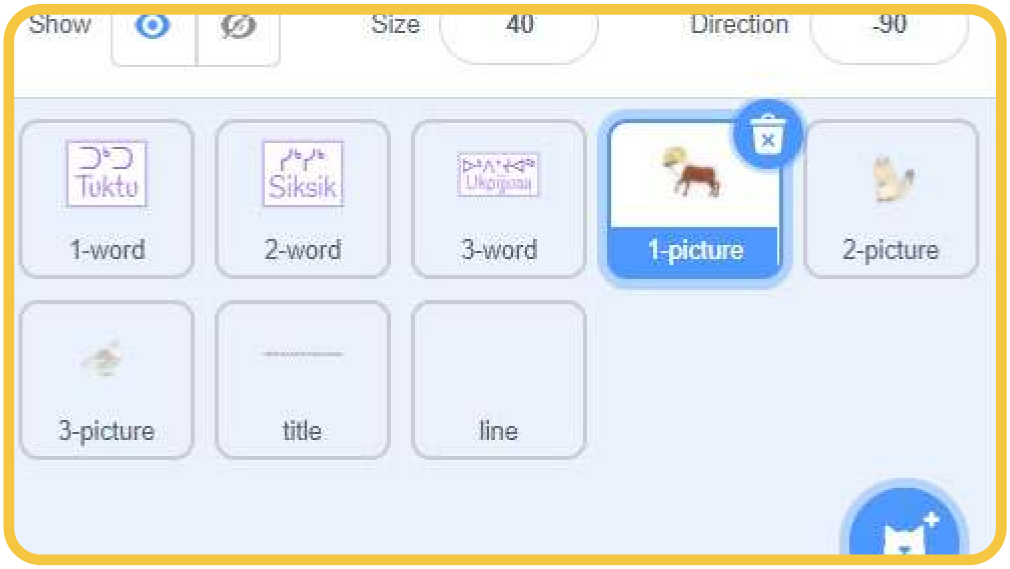
In this Scratch project, there are three sprites with Inuktitut words and three sprites with matching pictures. Students can click on a word then click on the animal they think it matches with, and the game will draw a line to connect the two. After the students guess all three matches, the game will draw coloured lines to show which are correct (a green line) and which were not (a red line). If the student missed any of the matches, the lines will erase themselves and they can try again. If all of the student's guesses were correct though, they win the game!

To help keep things organized, the names of the word and picture sprites have numbers in them to show which word should match with each picture: sprite "1-word" matches with "1-picture", "2-word" matches "2-picture", and "3-word" matches "3-picture".

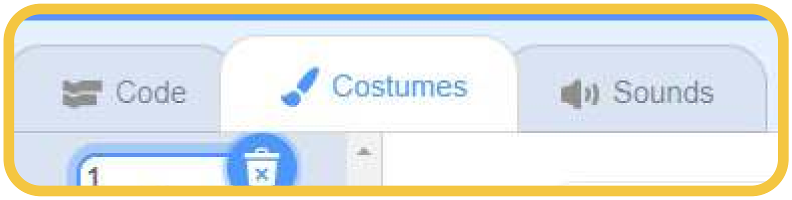
To make your own version of the game, you will need to edit the costumes of all three word sprites and all three picture sprites.

Editing a Picture Sprite

1. Select the picture sprite you want to edit



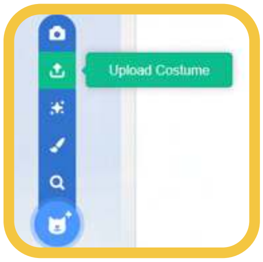
2. Open the Costumes tab



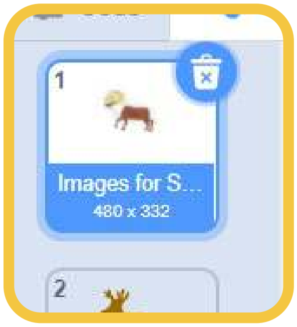
You can choose to upload an image you already have on your computer or draw your picture in Scratch. We have included an image resource from drawings made by Leo Karetak for sprites and photographs for backdrops taken by Jonah Oolayou which are available for you to use.

Upload an image you already have on your computer:

1. Move your cursor over the Choose a Costume button and select the Upload Costume option

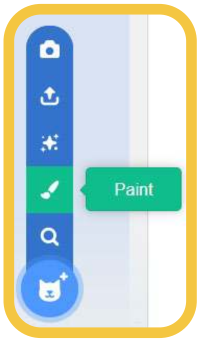


2. Navigate to the folder where the image is saved, select the image, then click the Open button
3. You can remove the other costume by selecting it and deleting it using the garbage button

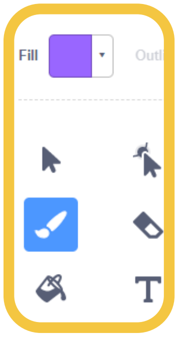


Draw your own picture in Scratch:

1. Move your cursor over the Choose a Costume button and select the Paint option



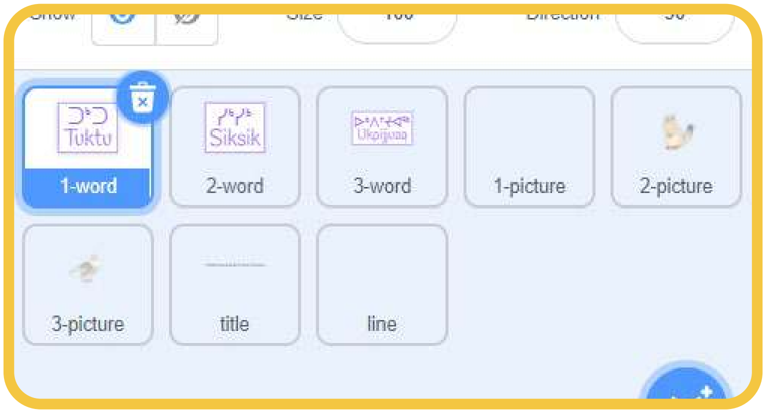
2. Select the Brush tool and choose a colour to paint with



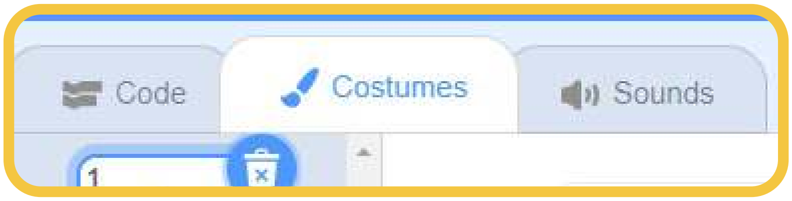
3. Click and drag in the costume window to paint your picture
4. Use the Fill tool if you want to fill a shape with a colour
5. Use the Line, Circle, or Rectangle tools to draw different shapes
6. Use the Eraser tool or Undo button if you make a mistake
7. You can remove the other costume by selecting it and deleting it using the garbage button

Editing a Word Sprite

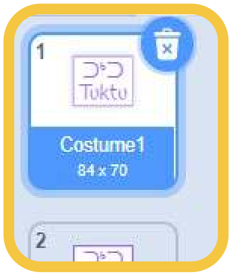
1. Select the word sprite you want to edit



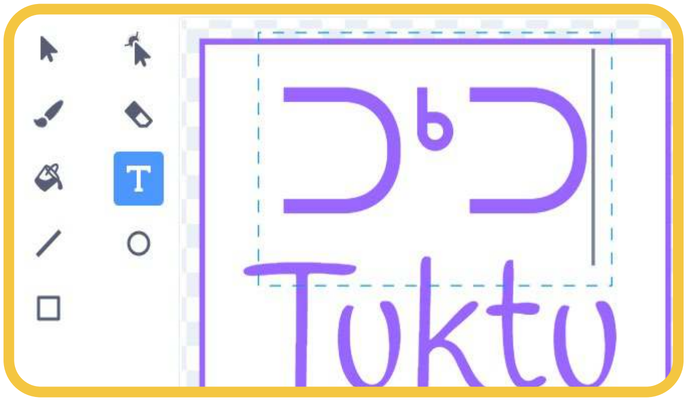
2. Open the Costumes tab



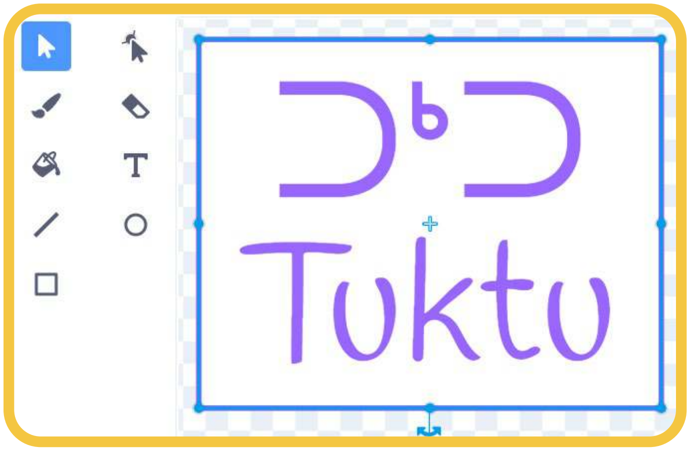
3. Select Costume1 from the list of costumes



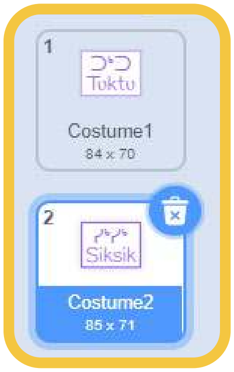
4. Select the Text tool then click on the syllabic word



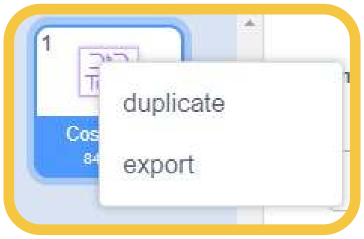
5. Use the backspace key on your keyboard to erase the text, then type in the new syllabic word
6. Do the same with the Roman Orthography version of the word
7. Select the Arrow tool then click on the rectangle behind the words
8. Move your cursor over the dot on the right side of the rectangle then click and drag to adjust the size of the rectangle



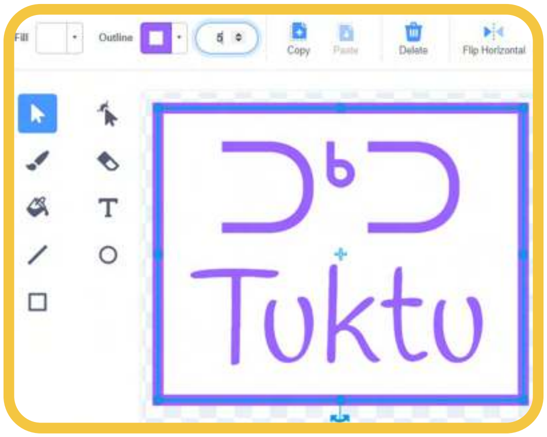
9. Now select Costume2 from the list of costumes and delete it using the garbage button



10. Right-click (or tap and hold on a touchscreen device) on Costume1 and select the Duplicate option to make a copy of it



11. Select the new Costume2
12. Select the Arrow tool then click on the rectangle behind the words
13. Make the outline of the rectangle thicker by changing the number to "5"



14. With both costumes now ready, click and drag the sprite in the project window to adjust its position

Why do the word sprites have two costumes?

You may have noticed that all of the word sprites have two costumes that are almost the same. The only difference between the two costumes is the outline of the rectangle is thicker in the second costume. Having two costumes lets the player know when a word sprite has been selected while they're playing the game.

When the green flag is clicked and the game starts, all three word sprites switch to the first costume. When one of the word sprites is clicked, it switches to the second costume that has the thicker rectangle outline, showing the player that it has been selected.

This is very common when designing User Interfaces (UI) on computers. When we move our cursor over something on the screen or click it, it will change visually in some way so we can see that something has happened. Try moving your cursor around the Scratch editor to see the visual changes as you hover over or click on different things.

CONCLUSION

You can use this guide to learn more about Scratch to modify the games that we have created and to learn how to create your own interactive story games. Students can also create their own games. These can be shared among classes, schools, and communities to create a shared bank of interactive Inuktitut language resources. We are more than happy to support you and your learners through your Scratch learning journey. Please reach out to us at **communityprograms@pinnguaq.com** to let us know how we can support you.

We are grateful to have partnered with Ilitaqsiniq, Nunavut Literacy to create this guide and these games. We would also love your feedback and suggestions for ways that we can improve the guide and games. **Thank you for piloting this program with us!**