

Universal Translator with Scratch 3.0

Participants Perspective

Today I learned how technology can be used to help communicate across language barriers **by** building a universal translator in Scratch 3.0.

Learning Goal

- Develop a user friendly interface in Scratch 3.0 to allow for easy input and output of information.
- Use Scratch 3.0's Google Translate module to translate between languages.

Logistics

- Suggested length: 1-2 hours
- Age range: Grade 5 or Older
- Additional requirements: Basic programming skills in Scratch (2.0 or 3.0), including the use of variables and control loops.

Part Number	Section Title	Group Size	Materials
Opening Hook	Say What?	Whole group	<ul style="list-style-type: none"> • Whiteboard or chart paper with markers • Digital device with access to Google Translate • Method of showing: <ul style="list-style-type: none"> ○ Google Translate, and ○ Video (YouTube) to class
Part 1	Build a translator	Partners	<ul style="list-style-type: none"> • Digital device with access to Scratch 3.0 and the internet • Scrap paper and pencil (Optional)
Part 2	Use a translator	2 sets of Partners	<ul style="list-style-type: none"> • Digital device with access to Scratch 3.0 and the Internet
Part 3	Closing discussion	Whole group	<ul style="list-style-type: none"> • Digital device to show a video (YouTube)

Safety Considerations

- **Laptops/Computers-** Electrocutation hazard from frayed or damaged cords and tripping over unsecured cords. These issues can be mitigated by securing all electrical cords, particularly

extension cords, reaching over the floor (using tape) and immediately unplug them after use. Cords are plugged in/unplugged by adults, or by participants under adult supervision.

- **Internet:** Participants may come across materials that is not appropriate (including but not limited to violent or sexual images, racist/sexist commentary, and so on). Participants may use websites unrelated to the activity (social media, email, etc). When participants are using computers, activities should be monitored. When possible, each computer should be pre-set with the desired website already loaded. Review expectations with participants before you start the activity and develop a plan that they should follow if they come across inappropriate content.
- **YouTube:** Participants may come across materials that is not appropriate (including but not limited to violent or sexual images, racist/sexist commentary, and so on). Pre-plan the YouTube viewing that will occur. We suggest using websites like TubeChop to clip YouTube videos to remove ads, and avoid the “suggested viewing” and comment sections of the usual YouTube sites.

Framework Connections

Toolsets	Skillsets	Mindsets
<p>Knowledge Developing a working user interface (UI) in Scratch using loops and other flow control devices.</p> <p>Utilize the Google translate functionality in Scratch 3.0</p>	<p>Digital Skills Coding and bug fixing in Scratch, including flow control and UI development</p> <p>Online communication across languages</p>	<p>Digital Intelligence Inclusive communication and access. Removing barriers by building simple tools.</p>
<p>Resources Scratch 3.0 (beta) or newer</p> <p>Google Translate</p>	<p>STEM Skills Problem solving using available tools</p> <p>Algorithm development and implementation</p>	<p>Computational Thinking Understanding how someone might interact with software.</p> <p>Logical sequencing and cyclical processes</p>
<p>Experiences Develop a functional and authentic tool to assist in pan-language communication.</p> <p>Computational thinking - Using flow control to process information in a logical manner, creating a “user friendly” experience</p>	<p>Essential Employability and Life Skills Problem Solving & solving communication challenges using available resources</p> <p>Creativity of design</p>	<p>Digital Action Digital citizenship, helping accessibility and opening communication channels</p>

Nuts and Bolts

Hook: Say What?

- Place some (or all) of the following foreign words on board:

Ano?	يقول ما؟	说什么？
Quid dicere?	ἵνα τί λέγει;	Sema nini?
	Hva sa du?	

- Ask the participants if they can identify what ALL of these phrases are saying? While some might know one, they will likely not be able to translate all of these. Despite this, [technology is getting very close to removing the barriers of language](#).
- Have participants play with Google Translate (<https://translate.google.ca/>) on their connected device:
 - First, let participants translate words that they would like to test out. Can they change the input or output languages? Do they know any words in other languages that they might be able to test out?
 - After a few minutes, ask the participants to try to identify what the sayings on the board are again.
 - Most should translate to “Say What?” or something close to that. [The key here is that the technology can help to get close to the correct translations, but human context is still needed to put things in the right order and fill in any gaps. This is still the hardest part of making a universal translator.](#)
- Watch the short clip on the “Babel fish” from the Hitchhiker's Guide to the Galaxy”
 - <http://www.tubechop.com/watch/10074232> or
 - <https://www.youtube.com/watch?v=YWqHkYtREAE>
 - While this is clearly science fiction, [it is possible to build a device that will at least help us talk to each other, using simple technologies](#). This is what the participants will be completing today.

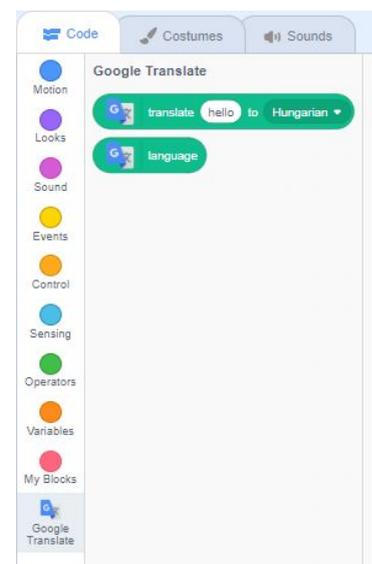
Part 1: Build a Translator

- Participants should pair up with a partner and load Scratch 3.0 on their connected device. This is currently (Nov 2018) available under BETA release at: <https://beta.scratch.mit.edu/>. Full release is expected early in 2019.
 - Note: If some participants are not familiar with Scratch, they should pair up with someone who has used the software before.
- We will start with testing out the Google Translate extension. Open the extension in the bottom left, by clicking the add extensions icon:



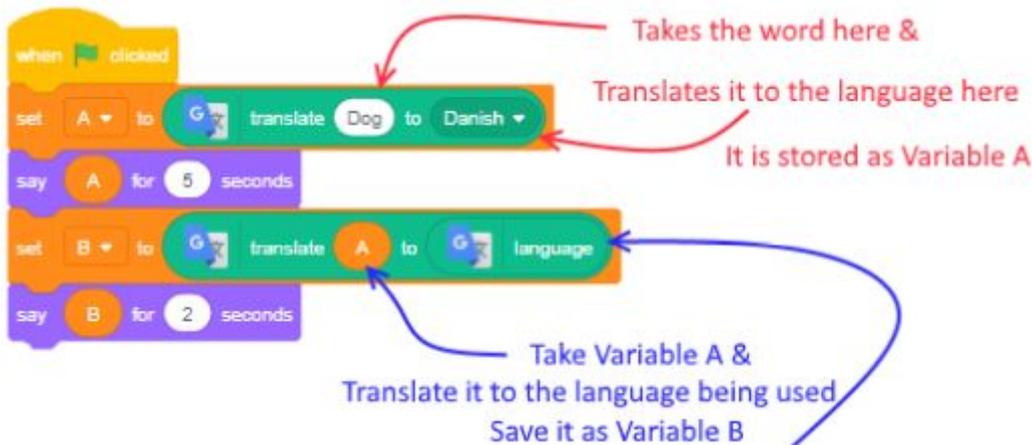
- This will add the Google Translate items to your option bar. There are 2 tools available to use here:

-  Allows you to translate the information in the white area (“hello”) into the language selected from the drop down green area (“zulu”)

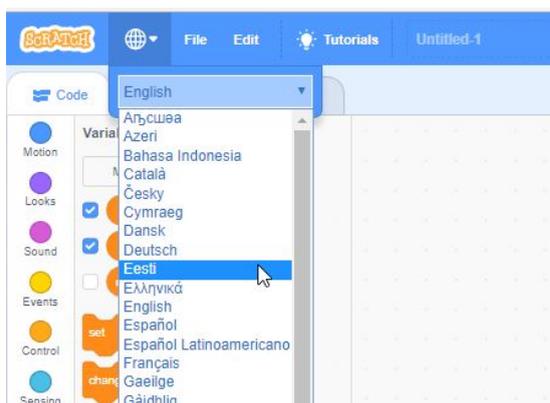


-  OR  Allows you to identify what language is being used by the person running the program.

- Next we are going to try these out with a very simple program that completes two tasks:
 - It makes the Scratch cat say the word “Dog” in another language of your choice and saves it as a variable.
 - Then it takes the translated word and translates that back into whatever language is being used for Scratch.
- Have the participants try it out themselves. After a few minutes, share what it should look like with those who are struggling. One example of this code is this (though many options are possible):



- Verify that all of the partnerships have been able to get this working. If some are done early, ask them to try changing the Scratch Language to something else to see if it will work for someone in another country (see image below). [Have the participants consider how this might be useful for someone who was going to another country or was new to Canada.](#)



- Now participants are going to embark on a challenge to make a universal translator. Read the following scenario to the participants:

- Someone new has just moved into your community and will be living very close to you. Your local community has volunteered you to be their guide to the community as you are almost the same age. The only challenge is that you do not both speak the same language. Luckily you are pretty good at Scratch! [Use your knowledge of Scratch and your problem solving skills to create a translator that will allow you to talk together.](#) Build a prototype with your partner to test out your skills. Note: you can pick a language for your new friend
- Tips:
 - **Reload Scratch again to start with a fresh program.** Switching the languages (as was done in the example) can sometimes cause issues with the variables.
 - Plan out on paper first. What might the program look like? [Planning first will lead to faster coding, better communication and problem solving, and will result in a better program overall.](#)
 - Consider how the program needs to be structured, such that each person gets a chance to ask and answer. Loops are a great way to do this!
- The following is sample code that will provide the basic functionality of the program. [As with any program there are numerous ways to do this and how participants approach the problem will help them to develop their problem solving and computational thinking skills.](#) Note that the appendix contains this code with comments about what each line is doing for reference.



- Participants should be encouraged to test out their translator and make changes as needed. Consider how you might want to test out the software. [This iterative process will help to build a better overall program, as it will help to fix software bugs and give the participants a chance to experience what the program might be like for the actual user.](#)
- Once the participants have working translators, they can move to the next step of sharing their translator.

Part 2: Use a Translator

- Pairs should now team up with other pairs to try out each others translators.
 - First have each pair try out the translator without and discussion of how it was built.
 - Next have each pair describe their program and how it was coded. [This description process helps to reinforce what aspects of the code are essential for each task.](#)
 - Then the pairs should discuss:
 - What did you like about their translator?
 - How was it different from how you built yours?
 - What is one improvement that you could suggest? How should this program be better? Consider how it would be used in the real world.

The act of both providing and receiving feedback helps to build a better understanding of the needs for the product and creates a better product in the end. It is also a helpful step in the learning process and in building strong communication skills.

- Pairs will now make amendments to their program based on both the feedback they received and what they learned from using and seeing the other pairs program. **Allowing the participants time to iterate on their idea will improve problem solving skills, critical thinking and develop a better product in the end.**
- Once the participants have completed their changes, they should test it out again.

Part 3: Closing Discussion

Discuss what the participants learned from this experience:

- Are they able to use the Google translate function in a Scratch program?
- **Do they think that their program would work to communicate with the new guest in their community?**
- **What was the most challenging aspect of building this program?**
- **What did they learn from trying out another group's code?**

Discuss again how this type of **technology is still not able to have a fluid conversation with two people, but this is a great start to be able to start and break down the communication barrier between languages. We can start to integrate technologies like this into simple programs so that these things become possible.**

Conclude by watching the following video showing Google's version of the same program working on their earbuds:

- <https://www.businessinsider.com/google-pixel-buds-language-translation-headphones-2017-11?jwsourc=cl>

Modifications and Extensions

Ways to make this activity more approachable:

- Groups of 4 work together to complete the design.
- Develop the code as a whole group with students as they follow along.
- Provide parts of the code, such as the flow control design.

Ways to make this activity more challenging:

- Have a participant or volunteer who can speak another language volunteer to test it out with the whole group.
- Iterate on the program to address the following questions:
 - Does it matter who speaks first?
 - What If I want to say a few things in a row?
 - How do you keep the information on the screen so it is there long enough to read?
- Modify the program to have an interesting background, multiple sprites that talk to each other or use sounds to signal the need for information to be input. Customize the program to make it unique.

Assessment and Evaluation

- Iterative design cycle and peer feedback on the code design.
- Group questions and discussion.

Credits, Kudos, Shoutouts

- Scratch 3.0 Programming language: <https://beta.scratch.mit.edu/>
- Google Translate: <https://translate.google.ca/>
- Babel Fish from: Adams, Douglas, 1952-2001. (1980). The hitchhiker's guide to the galaxy. New York :Harmony Books. Movie clip from 2005 version created by Touchstone Pictures and Spyglass Entertainment.
- Business Insider: Review of Google Pixel 2 Buds : <https://www.businessinsider.com/google-pixel-buds-language-translation-headphones-2017-11?jwsourc=cl>

Terms of Use

Prior to using this activity or parts thereof, you agree and understand that:

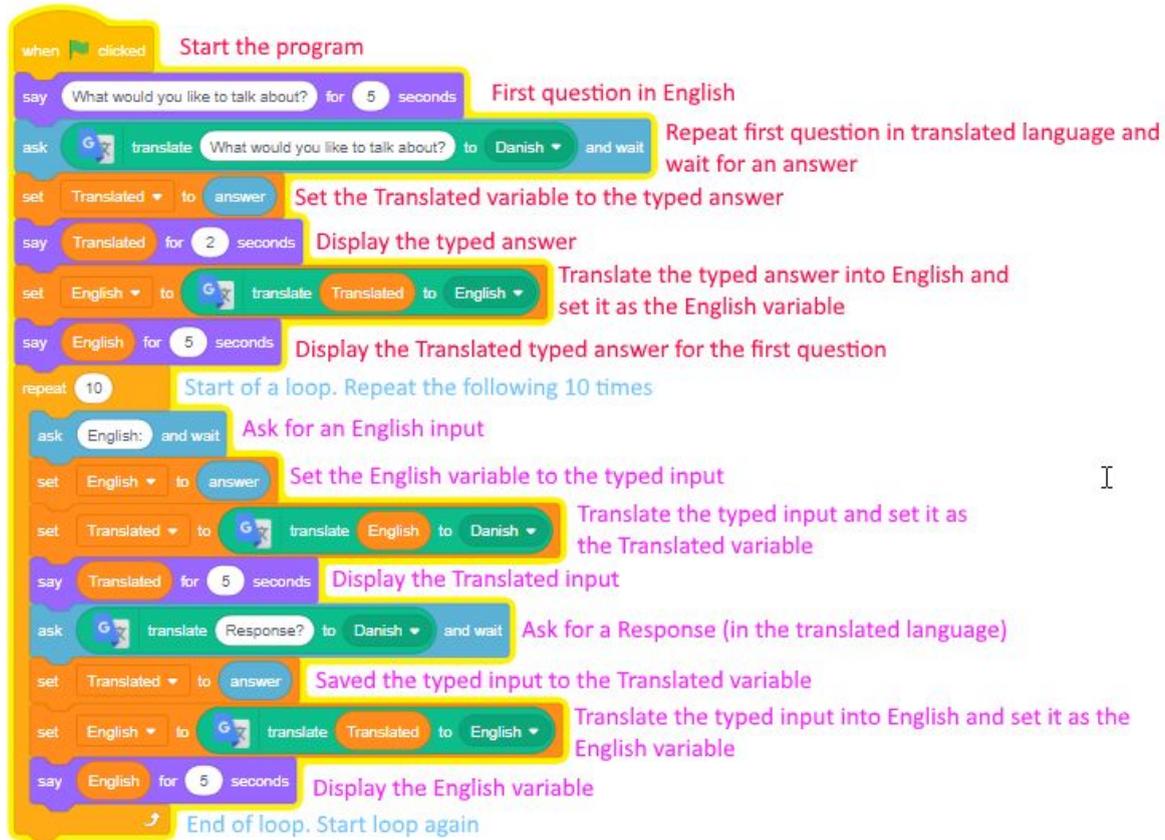
- It is your responsibility to review all aspects of this activity and ensure safety measures are in place for the protection of all involved parties.
- Any safety precautions contained in the “Safety Considerations” section of this write-up are not intended as a complete list or to replace your own safety review process.
- Actua shall not be responsible or liable for any damage that may occur due to your use of this content.
- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. For more information, please see <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

About Actua

Actua is Canada’s leading science, technology, engineering and mathematics (STEM) youth outreach network representing a growing network of university and college based members. Each year 250,000 young Canadians in over 500 communities nationwide are inspired through hands-on educational workshops, camps and community outreach initiatives. Actua focuses on the engagement of underrepresented youth through specialized programs for Indigenous youth, girls and young women, at-risk youth and youth living in Northern and remote communities. For more information, please visit us online at www.actua.ca and on social media: [Twitter](#), [Facebook](#), [Instagram](#) and [YouTube](#)!

Sample code with line-by-line comments:

See Universal_Transl.sb3 file for code



```
when clicked Start the program
say What would you like to talk about? for 5 seconds First question in English
ask translate What would you like to talk about? to Danish and wait Repeat first question in translated language and wait for an answer
set Translated to answer Set the Translated variable to the typed answer
say Translated for 2 seconds Display the typed answer
set English to translate Translated to English Translate the typed answer into English and set it as the English variable
say English for 5 seconds Display the Translated typed answer for the first question
repeat 10 Start of a loop. Repeat the following 10 times
  ask English: and wait Ask for an English input
  set English to answer Set the English variable to the typed input
  set Translated to translate English to Danish Translate the typed input and set it as the Translated variable
  say Translated for 5 seconds Display the Translated input
  ask translate Response? to Danish and wait Ask for a Response (in the translated language)
  set Translated to answer Saved the typed input to the Translated variable
  set English to translate Translated to English Translate the typed input into English and set it as the English variable
  say English for 5 seconds Display the English variable
End of loop. Start loop again
```