STIG^{CT}: The Science Teaching Computational Thinking Inquiry Group

A Community of Practice-Based Professional Development around CT Integrated Science Lessons for Pre-Service and In-Service teachers

Merijke Coenraad, Jandelyn Plane, Diane Jass Ketelhut, Lautaro Cabrera, Kelly Mills, and Heather Killen
University of Maryland, College Park

December 2020

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

This material is based upon work supported by the National Science Foundation under Grant No. 1639891.

Over the course of two years, our team of computer science, elementary science, and teacher educators partnered with more than 50 pre-service and in-service teachers to design, develop, and implement The Science Teaching Computational Thinking Inquiry Group (STIG^{CT}) using Design Based Research. Through two iterations, we identified best practices and key activities for professional development (PD) that support teachers to integrate computational thinking (CT) into elementary science lessons. In this guide, we provide activities used within our STIG^{CT} PD and the rationale behind each activity. In the second year of STIG^{CT}, when teachers went through these activities, over 80% of teachers successfully wrote and enacted a CT integrated science lesson for their elementary classroom.

Broadly, STIG^{CT} aims to support the development of CT integrated science experiences. It has four goals:

1. creating a community of teachers interested in CT,
2. providing PD about CT content, tool, and pedagogy to support integration,
3. building upon teachers' existing knowledge,
4. scaffolding teachers' design of integrated lesson plans for their classrooms.

Participants: Working toward these goals, STIG^{CT} brings together pre-service and in-service teachers with university researchers to work together and develop a community of practice. Many of the pre-service and in-service teachers were partnered as mentor teachers and pre-service student teachers within their classrooms, but this was not a requirement and un-partnered pre-service and in-service teachers participated.

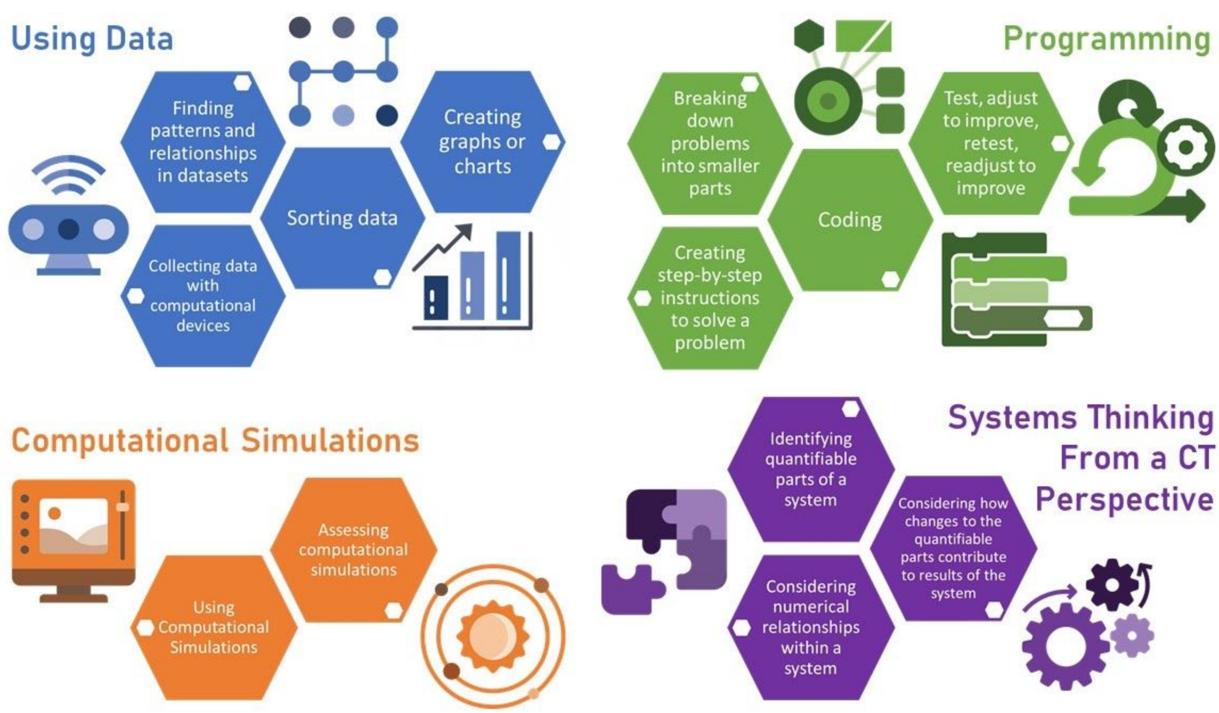
Prior CT Knowledge: Since the focus of our professional development was CT integration, all participants had prior knowledge of CT. This was ensured by our team collaborating with instructors of the pre-service science methods course (pre-service teachers) and hosting a starter course covering the corresponding material before STIG^{CT} began for in-service teachers (see our guide on Integrating CT into Elementary Science Methods Teacher Education Course).

Session Structure: Each STIG^{CT} session was based on one CT Practice (see Framework section below) and was divided into three parts: a presentation of the CT practice, participating in CT infused science activities from a student lens, and co-designing the beginning of a lesson plan, a "lesson seed," with grade-similar pre-service and in-service teachers and at least one facilitator.

We found it was essential to provide teachers with not only knowledge of CT and examples of CT, but specific practice in integrating CT into the science classroom and examples of this integration. We also found that the lesson seed planning time where teachers and facilitators co-design CT integrated science lessons provided an optimal opportunity for both generating lesson plans and assessing teachers' CT knowledge in order to correct misconceptions.

Timeframe: STIG^{CT} met monthly for four months in the Spring semester. Each meeting was 3 hours. The spring semester was chosen so that pre-service teachers could finish their elementary science methods course, held in the Fall, where CT was integrated into the activities and be in their full-time student teaching placement.

Framework (Ketelhut et al., 2019[1])**:** We used a framework for CT in elementary science specifically designed by our team. While we used a combination of a previously published framework and definition in the first year, we found that it was not specific enough to teachers' elementary science context. Our resulting framework is based on the work of Weintrop et al. (2016), but specifically focuses on the elementary school context. For more information on the framework, see "Our Framework"



Within this guide, you will find an overview of the goals and activities from STIG^{CT}. For each activity, there is a detailed description of how to run the activity, required materials, the rationale behind including that activity in STIG^{CT}, and our tips for success and future considerations. We hope the guide provides you with resources to support teachers as they integrate CT into elementary science.

---

[1] Ketelhut, D. J., Cabrera, L., McGinnis, R. J., Plane, J., Coenraad, M., Killen, H., & Mills, K. M. (2019). Exploring the Integration of computational Thinking into Preservice Elementary Science Teacher Education. *National Science Foundation STEM+C PI Meeting*. http://stemcsummit.edc.org/slides/DianeJass.pdf

STIG^{CT} Outline & Contents of the Guide

Main Topic	Goals	Activities
Programming Practices Agenda Presentation	Participants will: 1. Engage with and then reflect on how they used programming practices in science learning activities using Scratch and describe how programming practices are relevant to elementary science instruction at their grade levels and within their specific teaching contexts 2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate programming practices into the teaching of a science topic within their curriculum	1. CT Vignette 2. Programming with Conditionals: Using Scratch to Engage in Science Content 3. Lesson Seed Co-Design
Data Practices Agenda Presentation	Participants will: 1. Explore aspects of computational thinking in science lessons about the topic of weather (air masses/weather fronts). 2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate programming practices into the teaching of a science topic within their curriculum	1. CT Vignette 2. Weather Fronts Breakout Rooms a. Weather Fronts Breakout b. Scratch for Weather c. Weather Sensing micro:bits 2. Lesson Seed Co-Design
Systems Thinking through Computational Simulations Agenda Presentation	Participants will: 1. Explore aspects of computational thinking in computational simulations of physics topics. 2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate CT practices into the teaching of a science topic within their curriculum	1. CT Vignette 2. Tinkering with Examples of Modules/Simulations 3. Grade Specific Simulations 4. Stories from the Field: Lesson Plan Sharing 5. Lesson Seed Co-Design
CT Integrated Science Lesson Plans Agenda Presentation	Participants will: 1. Reflect on the success and limitations of their and their peers CT integrated science lessons. 2. Share a science learning activity integrating CT practices that was implemented in their classroom.	1. CT Vignette 2. Stories from the Field: Lesson Plan Sharing 3. Reflection and Discussion

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Additional Resources	Unplugged activities to provide additional computational thinking for students. When working with teachers, be sure to use these in conjunction with other activities and make explicit connections between the computing content and science integration.	1. Human Robot Activity 2. Synthesize a Story Jigsaw 3. Systems Thinking Challenge Stations 4. What's My Mystery STEM Career

STIG Session #1 - Programming Practices

Overview

The content focus of this session is the introduction of programming practices (via engagement in a science learning activity using Scratch) as a dimension of computational thinking (CT) that can be integrated into elementary science classrooms. An important aspect of this session will be to lay a foundation for the development of a community of practice between diverse groups of participants (in-service & preservice teachers; participants & facilitators) through informal conversation and formal collaboration in hands-on activities and guided discussions related to CT integration in science.

Participants will:

1. Engage with and then reflect on how they used programming practices in science learning activities using Scratch and describe how programming practices are relevant to elementary science instruction at their grade levels and within their specific teaching contexts
2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate programming practices into the teaching of a science topic within their curriculum

Timing	Procedures	Team members
One hour prior to the start of the session	**Set up** ● Refreshments ● Check-in table ● Session Presentation ● Scratch ready to go on Chromebooks	All Team Members
20 minutes prior to the start of the session	**Check in** ● Welcome participants and ask them to make a nametag ● Mark attendance for all participants	2 Team Members
5 min	**Welcome** Facilitator welcomes all participants and introduces the STIG with an emphasis on the "inquiry group" aspect and creating a community of practice together. Give overview of agenda	Main Facilitator
10 min	**Community Building** Participants get in two groups (can further subdivide groups depending on numbers): ● Pre-service Teachers	1 Team Member Facilitates, All Team Members Circulate

	• In-service Teachers Introduce yourself and share something interesting about your current classroom; Find something that you all have in common Small groups introduce themselves to the whole group	
15 min	**CT Vignette #1**	1 Team Member Facilitates
30 min	**Programming with Conditionals: Using Scratch to Engage in Science Content**	All team members circulate between groups
15 min	**Debrief and Connect to CT Practices** In advance of share-out, ask participants to consider what CT programming practices they (and their peers) used (show framework). Then give a few volunteers a few minutes to share their projects. Note: This is a discussion as a *whole group*, not just the teachers, so facilitators participate in the discussion as equals	1 Team Member Facilitates
10 min	**Break**	
10 min	**Introduce Lending Library** Give a brief presentation on your lending library - what's available and how to borrow. Highlight one or two items.	1 Team Member Facilitates
5 min	**Introduce Lesson Seed Task and Template** Explain that we just modeled a way of integrating programming with science topics in the elementary science curriculum. Now we would like them to come up with an activity that could engage students in CT while learning science. They will not be creating a full lesson plan, but a lesson seed (idea) that could be built out into a lesson plan Present lesson seed template.	1 Team Member Facilitates

5 min	**Introduce [Facebook Group]** Explain the purpose of the Facebook group: to will use to share resources and ideas during and between the sessions Today, we are asking them to summarize their lesson seed on the platform so their colleagues can benefit from the ideas discussed in their small group. Prompt them to reflect on the lesson in a discussion after they do the lesson in their class.	1 Team Member Facilitates
30 min	**Lesson Seed Work Time**	All Team Members
10 min	**Debrief** Each group briefly explains their lesson seed.	1 Team Member Facilitates
5 min	**Connecting to practice** Explain that between now and next time, we would like them to try out one of the lesson seed ideas in their classroom, and report back (via Facebook) on how it went. Can share any tips with others about adjustments you made or challenges you faced	Main Facilitator
10 min	**Stories from the Field** A couple of teachers share their stories from the field (brief accounts of how they've incorporated CT into their science teaching) after their methods course or CT introduction.	Main Facilitator
5 min	**Reflection and Closing** Participants complete written reflection in the Facebook Group. They will respond to the question, "After today, what questions do you have about computational thinking or how you could incorporate computational thinking into elementary science?"	1 Team Member Facilitates
	Post-session Stay around for a few minutes to answer any participant questions, particularly related to the lending library	At least 1-2 Team Members

STIG Session #2 - Data Practices

Overview

The content focus of this lesson is the integration of computational thinking in science lessons about the topic of weather, specifically weather fronts. Participants will participate in breakout sessions focusing on using computational devices to analyze and model data and CT data practices. During this session, continue the development of a community of practice during both the breakout sessions and through the lesson seed designing.

Participants will:

1. Explore aspects of computational thinking in science lessons about the topic of weather (air masses/weather fronts).
2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate programming practices into the teaching of a science topic within their curriculum

Timing	Procedures	Team members
One hour prior to the start of the session	Set up ● Refreshments ● Check-in table ● Session Presentation	All team members
20 minutes prior to the start of the session	**Check in** ● Welcome participants and ask them to make a nametag ● Mark attendance for all participants	2 Team Members
5 min	**Welcome**	Main Facilitator
5 min	**CT Vignette #2** Teachers can also start the vignette as they are arriving and getting seated.	1 Team Member Facilitates
1 hour 30 minutes (10 minute intro, 20 minute breakouts)	**Weather Fronts Breakout Rooms** Introduce the weather breakout rooms (or breakout groups if no extra rooms are available) and the routine for the day. Make sure teachers know they should work through these stations from a student lens. Before breaking out, ensure everyone knows their group and has access to a device and the internet. Teachers break into their individual rooms and rotate between groups. Two groups go to the Weather Fronts Breakout and one to each of Scratch for Weather and micro:bits. Groups rotate according	At least 1 Team Member per room

	to the schedule below. • Weather Fronts Breakout Room • Scratch for Weather • Weather Sensing micro:bits	
5 min	**Break**	
10 min	**Debrief Breakout groups** As a group, discuss: "How can these practices be used to understand other scientific concepts in your curriculum?"	1 Team Member Facilitates, All team members participate
30 min	**Lesson Seed Work Time**	All team members
5 minutes	**Lesson Seed Part 2 Reminder** Remind teachers they will need to build out one lesson seed (theirs or someone else's) into a full lesson plan and complete Part 2 before the last STIG session. Encourage teachers to post their developed lesson plans on Facebook prior to implementing to receive feedback from the group.	1 Team Member Facilitates
10 minutes	**Reflection and Closing** Reminder about lending library Participants complete a written reflection on a half sheet of paper. They will respond to the question: "After today, what questions do you have about computational thinking or how you could incorporate computational thinking into elementary science?"	Main Facilitator
	Post-session Stay around for a few minutes to answer any participant questions.	At least 1-2 Team Members

WEATHER FRONTS BREAKOUT SCHEDULE

<table>
<tr><th></th><th>Tornados</th><th>Hurricanes</th><th>Thunderstorms</th><th>Blizzards</th></tr>
<tr><td>Time period 1</td><td colspan="2" rowspan="2">Weather Fronts Breakout Room
- WHAT are weather fronts?
- WHEN are weather fronts?
- WHERE are weather fronts?</td><td>micro:bits</td><td>Scratch</td></tr>
<tr><td>Time period 2</td><td>Scratch</td><td>micro:bits</td></tr>
<tr><td>Time period 3</td><td>micro:bits</td><td>Scratch</td><td colspan="2" rowspan="2">Weather Fronts Breakout Room
- WHAT are weather fronts?
- WHEN are weather fronts?
- WHERE are weather fronts?</td></tr>
<tr><td>Time period 4</td><td>Scratch</td><td>micro:bits</td></tr>
</table>

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

STIG Session #3 - Systems Thinking through Computational Simulations

Overview

The content focus of this lesson is using computational simulations to explore complex systems, specifically physics systems. Participants will have the opportunity to explore various simulations designed for elementary-aged students and examine how computational thinking provides opportunities to understand unobservable science phenomena. The community of practice will continue to work together both in the CT and science integrated activities and the lesson seed design. Teachers will also have the opportunity to begin sharing their experiences integrating CT into their elementary science lessons.

Participants will:

1. Explore aspects of computational thinking in computational simulations of physics topics.
2. Collaboratively develop a science learning activity idea (lesson seed) that can be implemented in the classroom to integrate CT practices into the teaching of a science topic within their curriculum

Timing	Procedures	Team members
One hour prior to the start of the session	**Set up** ● Refreshments ● Check-in table ● Session Presentation	All team members
20 minutes prior to the start of the session	**Check in and direct to small groups** Welcome participants and direct them to the correct small group	2 Team Members
5 min	**Welcome** Give overview of the agenda and focus on systems thinking through models and simulations.	Main Facilitator
10 min	**CT Vignette #3** Teachers can also start the vignette as they are arriving and getting seated.	1 Team Member Facilitates
10 min	**Overview of computational models/simulations** Provide overview of models and simulations and the "big idea" from computational thinking illustrated in today's activities. There are topics in science that are hard to teach, hard to understand. **Think, Pair, Share:** In your opinion what is a topic in science that is difficult for your students to learn?	1 Team Member Facilitates

	Models are a tool to help us understand these concepts. Many times we use physical models (which have affordances) or physical simulations. • Models are representations of reality • Simulations are representations of processes • Computational modeling and simulations represent processes or systems based on their mathematical relationships **Think, Pair, Share:** How could a model or simulation help your students learn the topics you identified as hard to teach? Defining models and simulations • Review differences in affordances between physical models and computational models • Clarify that simulations that do not allow scenario changes based on variables are animations rather than simulations • Point out differences in using vs. creating (investigating questions model explanations) ○ Students can do either! **Think, Pair, Share:** What can students learn from each of these? Computational models have specific and different affordances that can make the hard to teach topics more accessible: • Easy to set up multiple/different scenarios • Illustrative of phenomena you cannot observe	
20 min	**Tinkering with Examples of Models/Simulations**	All team members
25 min	**Explore Grade Level Specific Simulations**	All team members
10 min	**Debrief CT Activities** Revisit the "big ideas" from computational thinking each activity illustrated. • Differences in affordances between physical models and computational models • Simulations that do not allow scenario changes - animation vs. simulation • Using vs. creating (investigating questions model explanations)	1 Team Member Facilitates

	Discuss the question, *"How can these practices be used to understand other scientific concepts in your classroom?"*	
5 min	**Break**	
30 min	**Stories from the Field** ● Teachers will share with the whole group about their experience implementing a CT infused lesson in their classroom. ● Potential emphasis: ○ Lesson seed adaptation ■ What lesson seed did you choose? ■ Did you use the lesson seed directly or did you develop or adapt? ■ Which CT practice did you include? ○ Reflection ■ Obstacles? ■ What went well? ■ Do you think this topic was a good fit for CT? ■ Do you wish you had included CT in another topic or lesson? ● Encouraging participation and questions from other teachers.	1 Team Member Facilitates: Teachers should be the main participants in this conversation
40 min	**Lesson Seed Work Time**	All team members
10 min	**Reflection and Closing** Reminder about Lesson Seed Part 2 - encourage teachers to post their developed lesson plans on Facebook prior to implementing to receive feedback from the group. Reminder about lending library Participants complete a written reflection on half sheet of paper. They will respond to the questions "After today, what questions do you have about computational thinking or how you could incorporate computational thinking into elementary science?"	1 Team Member Facilitates
	Post-session Stay around for a few minutes to answer any participant questions, particularly related to the lending library	At least 1-2 Team Members

STIG Session #4 - Reflection and Lesson Plan Sharing

Overview

In this session, participants will have the opportunity to share the lesson plans they created with their peers within the community of practice and reflect on the experience of being in STIG^{CT}. This session brings together the work done to create a community of practice between diverse groups of participants (returning, new, and pre-service); between participants and our team) by allowing them to share with each other and receive feedback from the community.

Participants will:

1. Reflect on the success and limitations of their and their peers CT integrated science lessons.
2. Share a science learning activity integrating CT practices that was implemented in their classroom.

Timing	**Procedures**	**Team members**
One hour prior to the start of the session	**Set up** • Refreshments • Check-in table • Session Presentation	All team members
20 minutes prior to the start of the session	**Check in and direct to break out groups** Welcome participants and direct them to the correct table to work with their small group during the session activities.	2 Team Members
5 min	**Welcome** Give overview of the agenda and focus on sharing their lesson plans and reflecting on the STIG^{CT} experience	Main Facilitator
10 min	**CT Vignette #4** Teachers can also start the vignette as they are arriving and getting seated.	1 Team Member Facilitates
60 min	**STIG Reflection Discussion**	All team members
10 min	**Optional: Time to Explore Lending Library Tools** When groups are finished with their reflection discussion, give everyone time to explore the lending library tools they might not have used during STIG^{CT} and ask any questions that they have. Advise participants as to whether tools will remain available to them now that STIG^{CT} is concluding.	At least 1-2 Team Members

10 min	**Break**	
60 Min	**Stories from the Field** Lesson Plan Sharing, Reflection, and Feedback	All team members
5 min	**Opportunities to stay connected** Discuss opportunities for STIG community members to remain connected. This could include: ● Planned follow-up work ● Continued online community space, such as the Facebook group ● Continued availability of the facilitation team	1 Team Member Facilitates
5 min	**Closing** Thank all participants for being part of the STIG community.	1 Team Member Facilitates
	Post-session Stay around for a few minutes to answer any participant questions, particularly related to the lending library	At least 1-2 Team Members

CT Vignettes

Summary:
At each STIG^{CT} session, teachers read through and respond to a CT vignette of a hypothetical science teaching scenario. Each vignette focuses on different CT practices and encourages teachers to think about what CT integrated science is and what it should look like. Teachers read through the scenario, determine whether they think the scenario represents CT integrated science, and explain how they make that determination.

Time: 5-15 minutes per session

Instructions:

1. (First session only): Review the CT framework that was introduced during participants' science methods course and/or the pre-course on CT that they completed.
2. Make sure each teacher has a copy of the CT framework handout.
3. Hand-out the CT vignette and instruct teachers to read through the vignette and complete the questions on their own.
    a. If teachers believe the vignette represents CT infused science, they explain how the lesson integrates CT and identify the CT practices in the vignette as is.
    b. If the teachers do not believe the vignette represents CT infused science, they describe how they would adjust the lesson described to include CT practices and which CT practices they would include.

Note:
These vignettes can also be used within the PD facilitator/researcher team. Before each STIG^{CT} session our team completed the vignettes individually and discussed our categorizations. We found these discussions fruitful to clarify our definitions of each CT practice and to prompt discussions of how our definitions of the CT practice differed based on our backgrounds. These conversations surfaced our perspectives and allowed us to enter the STIG^{CT} session prepared to address some of the misconceptions or definitions that participants voiced.

Resources:

- CT Framework
- CT Framework Handout
- CT Conceptualization Vignettes Bundle

Activity Rationale:
The definition of CT, what it is and what it isn't, is a difficult concept for people who have not been exposed to it before. There are also many definitions of CT available. Using the CT Vignettes provides teachers with scenarios where the boundaries of CT become the focus of the activity, making them an ideal way to encourage participants to think purposefully about how they define CT. The vignettes are not written to be clear-cut cases; their ambiguity forces teachers to make decisions about "what counts" as CT, which allows researchers to assess and guide teachers' understanding of CT.

Advice for Future Use:
When we implemented the vignettes, we did not discuss them with teachers, we used them only as an assessment of CT conceptualizations for research purposes. But, many teachers asked "how they did" on the vignettes and wanted to discuss and clarify whether the given scenarios indeed represented CT. If you are using the vignettes, we recommend making time for these conversations to help teachers (and yourselves) to clarify the definition of CT and what types of activities count as good CT integration examples. This is also an opportunity to clear up misconceptions that teachers have and tie back the CT integration demonstrated in the vignettes with the CT integration teachers are doing in their classrooms.

We also noticed that teachers had a tendency to assume that the vignettes were all good examples of CT in the science classroom, but that was not our intention. In fact, some of the vignettes were meant to contain no CT, and we were looking for teachers to identify the lack of CT in them. Make sure teachers know that these are not exemplars, they are designed to help think through the nuances of CT integration and how different practices can be integrated with varied success.

Community Facebook Group

Summary:
To provide an opportunity for teachers to communicate and collaborate outside of the STIG^{CT} sessions, we created a Facebook group teachers were welcome to join. This group acted as a location for teachers to post lesson seeds and lesson plans and remained open after STIG^{CT} for continued communication between teachers, if desired.

Instructions:

1. At the first STIG^{CT} session, explain the purpose of the Facebook group. Describe to teachers that the Facebook group is available to share resources and ideas during and between the sessions.
2. Explain that today and after each session, we are asking teachers to summarize their lesson seed on the platform so their colleagues can benefit from the ideas discussed in their small group.
3. Invite teachers to also share any experiences they have teaching CT integrated science and and questions they have that other teachers might be able to provide insights about.
4. Facilitators should visit the group regularly to encourage conversation and respond to teachers who are posting.

Activity Rationale:
Since the STIG is all about how teachers integrate CT into their classroom and questions would naturally arise between sessions, some type of on-line asynchronous communication is a good way to both allow teachers to question and share. It also allows everyone to give more ideas for the group. We used Facebook to try to carry the STIG to a platform available to teachers between sessions, but it is necessary to find a platform that provides the type of communication needed while also being an environment in which teachers are comfortable and active.

Advice for Future Use:
Collaboration and use of the Facebook group was varied. Teachers did post the content that we asked them to, but there was not much spontaneous communication. In the first year of STIG^{CT} we used a Piazza group and experienced an even greater rate of teachers not using the group. It would be worth considering other platforms and talking to teachers about where they are already collaborating with other teachers such as Slack or Discord.

Many teachers asked for a shared Google Drive folder where resources could be shared. While this does not have the same communication possibilities that other platforms have, it meets teachers where they are already doing work. Consider exploring how to create a community on Google Classroom or using shared docs in a Google Drive folder with resources for teachers.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Lesson Seed Planning

Summary:

At each STIG^{CT} session, teachers work with grade-similar peers and a facilitator to develop the beginnings of a CT-infused science lesson, a "lesson seed". During this time, teachers and facilitators work together in a Community of Practice model to incorporate the expertise of all individuals, especially the curricular and classroom knowledge of in-service teachers, technology knowledge of pre-service teachers, and CT knowledge of facilitators.

At some point in the 3rd or 4th month of STIG^{CT}, teachers select one lesson seed to develop into a full lesson plan and teach it to their students. Following the lesson implementation, teachers complete a lesson reflection about their experience.

Time:

Introduction to Lesson Seed Task and Template: 5 minutes
Lesson Seed Planning: about 30 minutes per session

Instructions:

Introduction to Lesson Seed Task and Template

1. Explain to teachers that they are going to be writing a lesson seed each session working with 2-4 other teachers and a facilitator. Groups will be based on grade-level in order to align content standards with CT-integrated activities.
2. Explain that the activities modeled in each session, such as those teachers just completed, are examples of how to integrate CT into science topics. If it is helpful, teachers can think about the technologies and tools introduced throughout the STIG^{CT} session and apply them to their own content and contexts. Teachers shouldn't copy what was done in STIG^{CT}, but these examples are a helpful scaffold for teachers as they think about their own context.
3. Show the lesson seed template to teachers.
4. Make sure teachers know that they are not creating a full lesson plan, but a lesson seed or idea that could be built out in different ways for their individual contexts.
5. Preview for teachers that they will adapt one lesson seed into a full lesson plan for their classroom before the end of the STIG^{CT}.

Lesson Seed Planning

1. Divide teachers into grade-similar groups of 2-4 teachers with one facilitator. Facilitators can move between 2 groups, if necessary.
2. Teachers select a content area/standard that they would like to focus on in their lesson seed. Encourage teachers to think about the science content that is coming in the next few weeks or months.
3. Once a science content context has been identified, teachers work together to brainstorm ways to integrate computational thinking and that content in a lesson that supports both authentic CT and inquiry science learning. Encourage teachers to think about how CT can enhance their science content and what CT understanding and science content knowledge partner well together.

4. Teachers write an objective for the lesson on the lesson seed planning template and a short description of the lesson. Then, teachers identify the CT practices that are present based on the CT Framework.
5. Teachers upload a picture of their lesson seed to the Facebook community group for others to see and consider.

Lesson Plan

1. Outside of the STIG^{CT} time, teachers identify one lesson seed to adapt into a full lesson plan. This can be a lesson seed they helped to write or one from another group.
2. Teachers implement their full lesson plan into their classroom.
3. Following the lesson implementation, teachers complete a Lesson Reflection (Lesson Seed Part 2) on the lesson plan in their classroom.

Lesson Seed Heuristic Discussion

1. Around the midpoint of STIG^{CT}, facilitate a heuristic discussion about how the lesson seed planning is going for teachers. Questions and further instructions can be found here: Lesson Seed Heuristic Discussion.

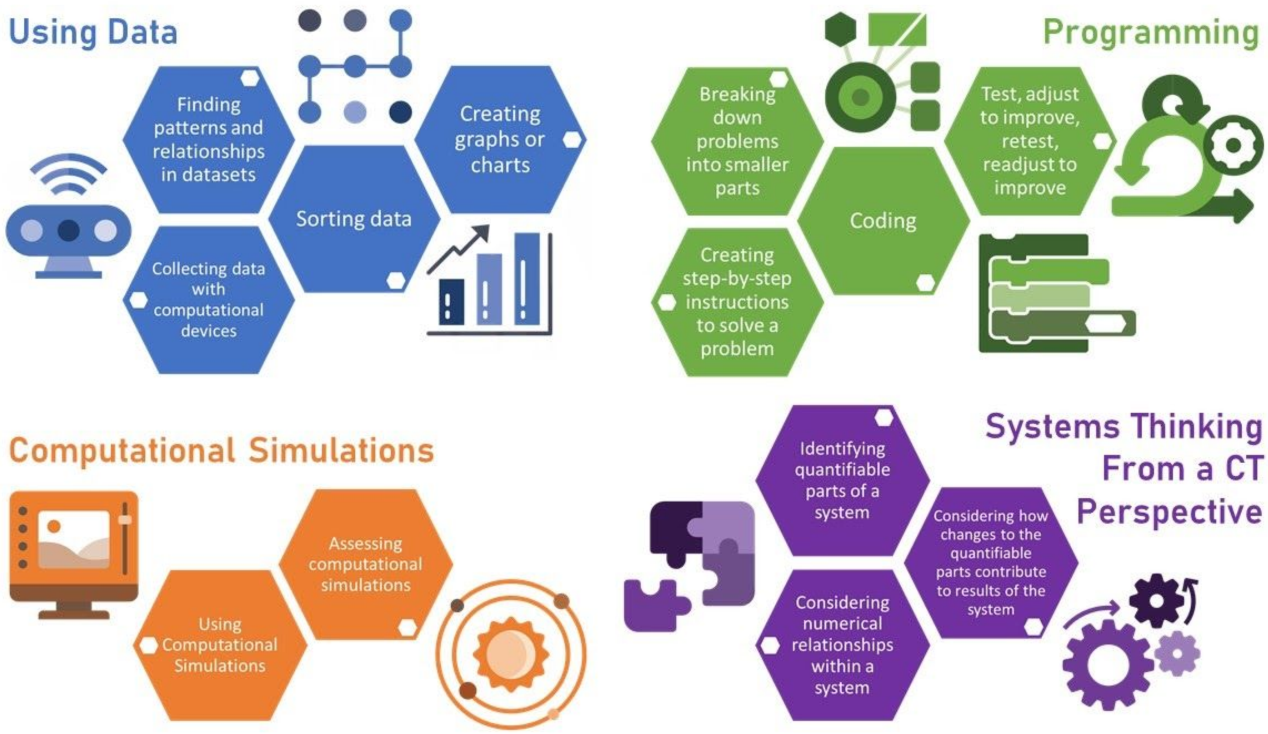
Resources:

- Lesson Seed Template
- Lesson Reflection (Lesson Seed Part 2)
- Lesson Seed Heuristic Discussion

Activity Rationale:
Actually applying CT to lessons teachers can take back to their classroom and use right away is the most important aspect of the STIG. Making a whole lesson plan is very time consuming and individual to the teachers. Instead we opted to have teachers work together to create lesson seeds that they could translate to their classrooms. The transition to creating lesson seeds at each session had several benefits. (1) The repeating of the process at each session gave them the opportunity to practice applying different CT concepts and skills beyond the ones that they are most comfortable (which would likely be the case if they were asked to create one complete lesson instead of lesson seeds). (2) Working as a group teachers are able to create lesson seeds and therefore support each other in that process but then each apply it differently based on the specific needs of their own classroom. (3) The seed gave us as facilitators enough information about how CT was being integrated so that discussions could help further refine and clarify the definition of CT in teachers' minds.

CT Framework

Summary:
Within STIG[CT] we utilized a framework of CT practices when defining computational thinking for teachers. This framework is based on Weintrop et al. (2016)[2], but has been modified to best fit within an elementary science context. The CT framework was used in all activities to help teachers to think about CT practices specifically relevant and appropriate to their teaching contexts.



Instructions:

1. Introduce the CT framework to pre-service teachers during their science methods course and to in-service teachers during the pre-course they complete before STIG[CT] begins.
2. In the first STIG[CT] session, remind teachers of the CT framework before teachers begin the CT vignette.
3. Each STIG[CT] session, make sure that all teachers have access to the CT framework handout.
4. Teachers use the CT framework when completing the CT vignette, planning their lesson seeds and lesson plans, and participating in activities during the STIG[CT] sessions.

Resources:

- CT Framework Handout

[2] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Framework Citation:

Ketelhut, D. J., Cabrera, L., McGinnis, R. J., Plane, J., Coenraad, M., Killen, H., & Mills, K. M. (2019). Exploring the Integration of computational Thinking into Preservice Elementary Science Teacher Education. *National Science Foundation STEM+C PI Meeting*. http://stemcsummit.edc.org/slides/DianeJass.pdf

Activity Rationale:

The simplified and applied CT framework gives teachers a concise tool to support them in integrating CT into their science lessons. This framework can also help researchers explain why model activities represent CT integration and identify how different components in lessons engage students in different CT practices. The use of the vignettes— and discussions around vignette responses— helps to align the interpretation of the framework so that we can move forward and know that terminology is being used consistently within the group.

Advice for Future Use:

The framework provided a strong foundation on which we could build teachers' understanding of CT and encourage the integration of CT practices within CT lessons. But, as we used the framework we noticed a tendency for teachers to identify activities as CT when those activities did not engage students with computational tools. For example, teachers identified typical science experiments or inquiry experiments as data practices even when data were collected and analyzed without computational tools and analysis. As you incorporate the framework into your work, we recommend the following.

- Stress a computational perspective (either plugged or unplugged) that focuses on whether the activities labeled as CT are preparing students to engage with computing to learn science.
- Use practices holistically, not independently.
- Explicitly discuss the difference between scientific inquiry and CT (which can be coupled with stressing the computational version of each practice).
- Pay attention to the language used to describe CT practices; the framework specifically uses less technical language, but this can lead to teachers identifying non-computational elements as a part of that practice.

Lending Library

Summary:
In order to make it easier for teachers to access novel technologies, we maintained a lending library of devices, microcontrollers, and robots. Teachers were able to check out any equipment from the lending library for use in their classrooms.

Time: N/A

Devices:
Our lending library included a variety of tools. We tried to provide a range of different tools that spanned appropriate age levels and different CT practices.

Our Lending Library included:

- Chromebooks
- iPads
- Code-a-Pillars
- BBC micro:bits
- Lego Mindstorm Robots
- KIBO Robots
- Finch Robots
- MakeBlock mBot
- 3-D Printer

Instructions:

1. Create a check out system to keep track of the tools and get feedback from teachers.
2. In the first session, present teachers with the Lending Library and share the tools that are available.
3. Remind teachers periodically throughout STIG^{CT} that the Lending Library is available for their use. This can be especially helpful when co-designing lesson seeds because it expands the possibilities of how CT can be integrated in teachers' classrooms.

Activity Rationale:
Not all schools have access to computational devices and this can create and perpetuate disparities between students in different buildings or districts. Even typical devices like iPads or Chromebooks could be hard for some teachers to access. Limited access leads to teachers either not being able to integrate CT into their classrooms or only using unplugged activities. By creating a Lending Library, we were able to help teachers overcome the barrier of access to technologies to integrate CT in new ways within their classrooms.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Lesson Seed Heuristic Discussion

Summary:
About halfway through STIG^{CT}, use this short activity to better understand how teachers are feeling about creating lesson seeds and where you can adjust the process to help them develop CT-integrated lesson plans.

Note: If you would like to keep this as data, consider having teachers record their answers on a worksheet or Google Document or audio recording the conversation.

Time: 10 minutes

Instructions:

1. Break teachers into their lesson seed groups for the day.
2. Begin the lesson seed planning time with a short discussion of these questions. Facilitators should take note of suggestions that teachers make and ways that the planning time could be improved for the remainder of STIG^{CT}.
3. Present the following questions on the board:
    a. How do you get started with the lesson seed?
    b. How have different group members (in-service teachers, pre-service teachers, CT team members) contributed to your lesson seed planning?
    c. What has been challenging about lesson seed planning?
    d. What has been helpful?
    e. What resources would be even more helpful?
4. Conclude the discussion with teachers' current impressions of CT integration in science.
    a. Are there science concepts that work better without CT?
    b. Why did you choose this particular CT practice to integrate in this science?
5. Once the group has finished their discussion, they should begin their lesson seed planning.

Activity Rationale:
Because we focused on creating a Community of Practice, we felt that it was important to partner with teachers throughout STIG^{CT}. One way that we did that was by getting feedback about activities and making adjustments as needed. Teachers appreciated the opportunity to talk with one another and with us about how STIG^{CT} was going and their experiences. The feedback also helped us to get an idea of teachers' confidence and how prepared they felt to present their CT-integrated lesson. This allowed us to provide additional support as they planned the lessons they would present to their classroom.

Programming with Conditionals: Using Scratch to Engage in Science Content

Summary:
Teachers receive an incomplete Scratch project using conditionals. When completed, the project will use conditionals to determine which animal a person is thinking of in 5 questions or less. The teachers must select questions that will narrow down the animals and use Scratch to complete the program using conditionals and the ask & answer blocks.

Time: 45 minutes

Instructions:

1. Introduce Scratch and the basics of block-based programming (if teachers are not previously familiar with the platform)
2. Show Scratch If-Then-Else and Ask Question blocks and explain their functionality.
3. Participants work in small groups:
   - You are an elementary student learning about animal classification in your class
   - Your teacher asked you to help the Scratch cat identify the animals by asking yes/no questions about them
   - You have 30 minutes to come up with a way for Scratch to identify which animal you're talking about. Try to do it in 5 or fewer questions.
   - Use this template to start: https://scratch.mit.edu/projects/279841436/
   - You will need to add code to the Scratch Cat sprite and different code to each of the animals.
4. Once most of the teachers have finished, debrief the activity:
   - In advance of share-out, ask participants to consider what CT programming practices they (and their peers) used.
   - Then, give a few volunteers a few minutes to share their projects.
   - In large group discussion:
     - What CT programming practices were used?
     - How did engaging in those CT practices influence your understanding of the science concepts?
     - How can programming be used to understand other scientific concepts?
     - How can we use other CT practices to understand animal categorization

   Note: This is a discussion as a whole group, not just the teachers, so facilitators participate in the discussion as equals

Activity Rationale:
Learning conditionals in programming is important both in the fact that it is an essential construct in any programming language and because it helps students tie programming to something they have been doing extensively in both science and life. The idea of categorizing items by different characteristics is something taught from a very young age and this lesson extends that practice from being done by the human mind directly to how to tell the computer to make those decisions. By using Scratch as the platform, this lesson is also able to introduce the object oriented paradigm of programming. Each sprite is given its set of directions of what it should do (object oriented paradigm) when a certain choice is made by the user. This

exercise is easy for the new programmer to understand because there are only two options the sprite has (being visible or not). It also is good for reinforcing the lesson because choices must be made for several different sprites in very similar ways.

Advice for Future Use:
Many of our teachers were unfamiliar with Scratch and found the platform to be slightly overwhelming, especially when paired with their belief that programming was one of the hardest CT practices. Make sure to provide ample time for teachers to work and experiment and just-in-time help by team members to support teachers.

Consider making the following adjustments:

- Identify the blocks teachers will need to use and add them to the script area of the Scratch project without connecting them to scripts so teachers can use a subset of the available blocks.
- Determine which teachers are familiar with Scratch before the activity and pair experienced and inexperienced teachers to support newcomers to Scratch.
- Create a guide that prompts teachers in small steps without giving them the exact code they will need to complete the project.
- Create a smaller version of the project or a smaller example and work up to this!

Scratch for Weather

Summary:
Teachers create a simulation showing that a weather event happens when two fronts interact using Scratch. Using sprites of their choice (either shapes they drew themselves or a sprite from the Scratch Library), they created a scene with two fronts and at least one other sprite representing a weather event (e.g. rain, lightning). This activity gives teachers an additional opportunity to practice and gain confidence with Scratch following the first programming introduction and demonstrates how students can use Scratch to model scientific phenomena such as weather.

Time: 20 minutes

Instructions:

1. Demonstrate a weather fronts Scratch project: https://scratch.mit.edu/projects/293676347. If possible, avoid showing teachers the code to limit the tendency to copy the example.
    a. When Green Flag Clicked - Initialize
    b. When space key pressed - Fronts move and rain appears when they touch
    c. When up arrow pressed - Warm front moves above cold front (demonstrate more realistic action of the fronts)
    d. When m key pressed - Fronts do not touch, so no weather event occurs
    e. When r key pressed - Fronts move and touch. When the rain appears it moves as if it is falling
    f. When s key pressed - The fronts move together slowly
2. Discuss other potential weather events that could occur:
    a. Lightning (sprite in the Scratch Library)
    b. Snow (if the weather is cold)
3. Hand out the Creating Simulations in Scratch handout and encourage teachers to try out creating their own animation modeling weather fronts.
4. Have the teachers share their projects to a studio so they can see the projects that others have created.

Resources:

- Creating Simulations in Scratch Handout

Activity Rationale:
This activity gives the participants an introduction to some programming basics while also showing them how easy it is to create a basic simulation. It provides teachers with a model of how programming activities can be used to support science learning. The balance of art and design creativity with the programming is a gentle way to introduce programming and simulations.

Advice for Future Use:
While some teachers were still overwhelmed by Scratch, this activity helped to demonstrate a simpler use of the programming environment (as compared to the conditionals programming activity). Teachers were very excited by the ability to create their own costumes and backdrops and some teachers went above and beyond with their coding.

A few things occurred spontaneously during our sessions that we would recommend that others try:

- Limit the amount of time teachers spend on the art aspects of the project to make sure they have time to code.
- Especially for tentative teachers, partner mentor teachers with their student teachers to support each other in the coding. We found that the pre-service teachers in the workshop often took on the role of technology expert when partnered with in-service teachers.
- Set small, achievable goals. This project lends itself to a few steps and extensions, such as:
    - Select assets for the project (sprites and backdrops)
    - Make one weather front sprite move while the other is stationary
    - Program a weather event to occur when the two sprites touch
    - Synchronize the movement of both weather front sprites
    - Allow the user to input a season and allow for different weather events and/or assets depending on the selected season

Weather Fronts Breakout Room

Summary:
Teachers will explore weather data and how to teach about weather fronts. Using https://weather.com/ and spreadsheets, teachers will examine visual and numerical data to make predictions about weather trends and potential weather events. Teachers will conclude each activity by reflecting on how CT can affect how they think about and experience data analysis.

Time: 40 minutes
This station will begin with two groups together for a 10 minute large group activity. Then, the two groups will split apart for 15 minute stations before rotating to the next room.

Instructions:

WHAT are Weather Fronts?

1. Question teachers about what they would do to teach a subject they are not knowledgeable of.
2. Give teachers the following scenario. Teachers spend 5 minutes learning about Weather Fronts on their own or with a peer.
    Scenario: You are challenged to teach their class a topic about something you have no idea about: Weather fronts! You have five minutes to teach yourself enough about this topic to teach your class.
3. Discuss: What did you do? What were you able to find in the 5 minutes?
4. Review what the CT Team found:
    a. Animated characters: https://www.youtube.com/watch?v=RD-2dvaG4UY
    b. Boring science video: https://www.youtube.com/watch?v=o4lg8UfY5DM
    c. Hot/cold water demonstration: https://www.youtube.com/watch?v=bN7E6FCuMbY

Breakout Groups

WHERE are the fronts?	**WHERE are the fronts?**
1. Distribute the WHERE are the Weather Fronts Handout. 2. Teachers go to weather.com and view temperature maps. 3. After exploring the temperature maps, teachers predict whether they think that weather events are going to happen based on where the weather fronts are located. 4. Teachers check their predictions by viewing a radar map. 5. Lead teachers to reflect: In this activity, did CT allow you to do something with science that you may not have been able to do otherwise? Explain.	1. Distribute the WHEN Were the Weather Fronts Handout 2. Teachers access the WHEN Were the Weather Fronts Data (Note: This data can be updated for your location and the timing of your PD) 3. Teachers examine the data and answer the questions on the handout to notice the extremes in the data. 4. Teachers experiment with using the spreadsheet software to analyze and notice trends in the data. 5. Lead teachers to reflect: In this activity, did CT allow you to do something with science that you may not have been able to do otherwise? Explain.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Resources:

- WHERE are the Fronts Handout
- WHEN Were the Weather Fronts Handout
- WHEN Were the Weather Fronts Data

Activity Rationale:

This lesson is important because it exposes teachers to readily accessible large data sets available online and because of the processes discussed how data can be analyzed on a basic level. The large group activity of finding what you can about a topic they were not ready to teach shows how quickly basic information can be found, understood, and vetted for reliability (by multiple supporting/refuting sites). By performing these searches, teachers can be more confident about teaching a lesser known topic because they can find student-appropriate information about it.

In the first breakout room, this is reinforced by predicting and then checking their prediction by comparing the temperature map with the radar map. This type of activity will help teachers support students to see that they need to analyze data given by one source (rather than just view it) and then they need to support their conclusions through other sources.

In the second breakout room, the basics of how to read and analyze data in a spreadsheet is the emphasis. One side benefit of this activity is to show that large sets of data are not intimidating because it is easy to find basic items such as extremes and patterns. This is important because when presented with a large set of data, some people are overwhelmed because they feel it is the numbers that they need to understand and absorb rather than characteristics of those numbers.

Weather Sensing micro:bits

Summary:
Teachers program BBC micro:bits to have different visual responses depending on the temperature or light level. Using block-based programming and the micro:bit's built in sensors, teachers develop a program that flashes when the micro:bit senses a certain temperature (e.g. freezing, dramatic temperature change) or when the light reaches a certain level. This activity demonstrates one of the tools available in the lending library and models how satellites and drones capture or respond to weather data. Teachers are also able to see how easy it is to have equipment like this promote scientific investigation and inquiry learning.

Time: 20 minutes

Instructions:

1. Demonstrate the micro:bit's temperature and light sensors using a pre-written program
    a. Avoid showing teachers the code but rather the output on the micro:bit
2. Explain to teachers that weather satellites and drones use sensors to capture or respond to weather data. These sensors are similar to those available on the micro:bit.
3. The Measuring Light and Temperature with the micro:bit guide walks teachers through basic programming in micro:bit Make Code and shows them how to send data back to the computer.
4. Teachers discuss how micro:bit sensors could be used to answer science questions about light and temperature through continuous data collection.
5. Teachers pick a question and use the micro:bit Make Code editor to write basic programs for the micro:bit that sense temperature or light change.

Resources:

- Measuring Light and Temperature with the micro:bit

Activity Rationale:
These activities show teachers how little programming is needed to use basic data processing tools and how data can be collected through computational devices. In doing the exercise, teachers were rewarded with an attainable working product at the end of the activity. The teachers were provided with a template project (just like students would be) so that they could create their product in a short period of time.

Grade Specific Simulations

Summary:
Teachers explore available simulations to determine potential simulations that align with their content standards and are appropriate for their students. As teachers are exploring, they reflect on whether what they have discovered online is truly a simulation and how the simulation can provide opportunities for student learning.

Time: 20 minutes

Instructions:

1. Break teachers into grade-similar groups. These can either be teachers' lesson seed groups or other teachers who teach within their grade band.
    a. Facilitators join each group or move between designated groups to offer additional thoughts and answer questions.
2. Teachers visit http://new.learningscience.org/wp/ to explore a variety of simulations that are available for use across grade levels.
3. Teachers explore together to find simulations they might use in their classrooms based on content and age-appropriateness.
4. As teachers are identifying simulations, discuss:
    a. If and how does each simulation provide opportunities for student learning?
    b. If and how are the simulations you are finding aligned to our criteria of a simulation (variables that can be manipulated, dynamic changes represented based on selected variables)?
5. Teachers complete a written reflection:
    a. Which simulation(s) did you explore? Do they provide opportunities for children to engage in science learning that they could not with a non-computational tool (such as a physical model)? Explain.

Resources:

- Exploring Computational Models and Simulations handout

Activity Rationale:
This activity has two different purposes. The first is to have teachers discuss the integration of simulations into their grade specific content. The second is to show them how easy it is to find simulations on different content that are already available. Their search for simulations, testing of those applications, and their analysis about if that simulation is appropriate for their age group and content all give them guided and group supported practice of what they would need to do to integrate these simulations into their classroom.

Tinkering with Examples of Models/Simulations

Summary:
Teachers explore examples of models and simulations with grade-similar peers to become familiar with using models and simulations and begin brainstorming how models and simulations could be used to teach phenomena that children are unable to observe. Teachers complete the activity with a discussion of how the provided simulations provide opportunities for students to engage in science and computational thinking.

Time: 20 minutes

Instructions:

1. Virtually distribute the Exploring Computational Models and Simulations handout
2. Break teachers into grade-similar groups. These can either be teachers' lesson seed groups or other teachers who teach within their grade band.
   a. Facilitators join each group or move between designated groups to offer additional thoughts and answer questions.
3. Provide teachers with the two simulations selected to demonstrate how simulations can illustrate phenomena that children are not able to observe:
   a. EarthViewer - Simulation to explore changes to the Earth over time (Scale - Time)
   b. Gravity and Orbits - Simulation to explore interactions between the Sun, Earth, and Moon under different conditions (Scale - Size)
4. As teachers are exploring the simulations, discuss:
   a. How do these simulations allow students to explore phenomena that they are unable to observe?
   b. How could a simulation such as this be used with your students?
5. Bring the group back together for a full-group discussion:
   a. If and how do these simulations provide opportunities for children to engage in science learning?
   b. If and how do these simulations provide opportunities for children to engage in computational thinking?

Resources:

- Exploring Computational Models and Simulations handout

Activity Rationale:
This activity provides teachers with the opportunity to consider different computer simulations available with the emphasis on grade specific integration of selected simulations into their curriculum. These specific simulations were chosen to demonstrate to teachers the affordances of simulations. They simulate things that can't be seen but also are difficult to picture in your mind without a model or simulation. The work with peers within the grade bands allows the teachers to take the content of that model but specifically look at it in terms of integration into the science topic for their classroom.

Advice for Future Use:
Teachers of students in lower grades (particularly 1st-3rd grade) struggled to make these simulations relevant to their students because they felt they were too difficult for the students to use on their own. When working with teachers in this grade-band, consider how to support teachers to see ways to make these simulations applicable to their students including a student-led full group exploration or a teacher-led station activity where experimentation could be performed in a small-group setting to provide greater student support.

STIG^{CT} Reflection and Discussion

Summary:
Teachers with both peers and STIG facilitators have the opportunity to reflect on: (1) the STIG experience, (2) integrating CT into their science classrooms, and (3) how they want to move forward. This reflection allows teachers to continue learning from the community of practice as they think about the process of increasing CT integration in their classrooms moving forward.

Note: These questions were developed by our team for the purpose of gathering research data on STIG^{CT} and teacher experiences. If STIG^{CT} is being used as a research study, this is an opportunity for researchers to hold focus groups that allow for both research data collection about experiences and teachers to have an open conversation about STIG^{CT} and integrating CT into elementary science. If STIG^{CT} is not being used as a research study, consider adjusting these questions to provide a more open reflection experience or omit this activity.

Time: 60 minutes

Instructions:

1. Separate teachers into small groups with a facilitator. Groupings could be randomly selected or by grade level, teaching role (in-service vs. pre-service), or district.
2. Facilitators lead a discussion with teachers about their experiences. Some questions to consider are:
    a. How have the learning experiences in the STIG influenced your ideas about CT in the classroom? What was useful? What could be changed to make it more useful?
    b. How comfortable are you integrating CT in an elementary science lesson?
    c. Besides the one required lesson, have you integrated CT in your classroom since beginning the STIG?
        i. If yes,
            1. How often have you integrated CT into your science classroom?
            2. How can CT change the way your students engage in science learning?
            3. Did you also integrate CT in other subjects? If so, which ones and how often?
            4. Why did you integrate CT?
            5. What were the challenges of integrating CT?
            6. Describe how your students reacted to the integration of CT.
            7. Were there CT practices that you were more likely to integrate? If so, which ones?
        ii. If no,
            1. Why have you not integrated CT?
            2. What are the challenges to integrating CT in your classroom?
    d. Are there some CT practices that are less challenging than others for elementary learners?
    e. (Pre-service teachers, only) Have you had a discussion about CT with your mentor teacher? Do they support the integration of CT? In what ways?
    f. Will you continue to integrate CT into science in your classroom? If so, how? What supports would be helpful to you as you continue this integration?

Activity Rationale:
This activity gives teachers a final opportunity to be in community with each other (at least in person) and learn from the thoughts and experiences of their peers. It helps the facilitators to gauge level of understanding and comfort and it gives all of the participants a structured way to do self reflection of what they have been doing in the STIG. Since it is a discussion about CT in general rather than any specific aspect, the variations in the definitions also emerge during the discussion and can be addressed so that all participants are using the same terms for the same things moving forward.

Stories from the Field

Summary:
Teachers share their experiences implementing a computational thinking integrated science lesson within their classrooms. As they share, teachers focus on their students' engagement in computational thinking and how the integration of computational thinking influenced the students' science learning.

Note: Stories from the Field can be used throughout the STIG^{CT}. If you have experienced teachers, it could be helpful to have them share early in the experience. As well, having a few teachers share during the second to last session gives everyone as an example of how lessons had gone in classrooms was a helpful model for teachers thinking about presenting their lessons and for those who had not yet taught their CT integrated lesson.

Time:
2nd to last STIG^{CT} Session: 30 minutes
Last STIG^{CT} Session: 60 minutes

Instructions:
Second to last STIG Session
1. Invite a few teachers (2-3) to share their experiences implementing a computational thinking integrated science lesson with the entire STIG^{CT} group. This provides an example for teachers who have not yet implemented their lesson and allow for a full group conversation. Presentations should focus on:
    a. What lesson did you teach?
    b. Did you use a lesson seed to create this lesson? If so, did you use it directly or adapt it?
    c. What CT practices did your lesson include?
    d. What obstacles did you or your students face?
    e. What went well?
    f. Do you think this science topic was a good fit for CT?
    g. Any other reflections or ideas that you have based on your experiences?
2. Provide time for other teachers to ask questions and for discussion between the teachers. Encourage discussion between teachers by seeding questions and allowing questions from the facilitators.

Final STIG Session
1. Prior to sharing, teachers should complete the first part of the Lesson Reflection
2. Break teachers into small groups of around 6 teachers and at least one facilitator. Group size will vary depending on how many teachers participate in STIG^{CT}, but should balance allowing everyone to share while also allowing teachers to hear about a variety of different CT integrated lesson plans.
3. Give each teacher around 5-10 minutes to share their lesson and provide a brief reflection. Presentations should focus on:
    a. What lesson did you teach?
    b. Did you use a lesson seed to create this lesson? If so, did you use it directly or adapt it?

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

c. What CT practices did your lesson include?
d. What obstacles did you or your students face?
e. What went well?
f. Do you think this science topic was a good fit for CT?
g. Any other reflections or ideas that you have based on your experiences?
4. Allow other teachers to ask questions about the lesson and discuss how a similar lesson could be integrated into their classroom context and what adjustments might need to be made.
5. Teachers complete the final part of the Lesson Reflection

Resources:

- Lesson Reflection

Activity Rationale:
Being able to share how CT was integrated into an actual lesson is very important to the STIG structure. If the integration of CT into lessons is kept on the abstract of looking at different things they could do, the participants don't gain enough confidence in how it can be integrated into the reality of their classroom. Discussing the challenges faced also raises confidence because they see various ways those challenges can be addressed before they are in the moment of stress when those challenges need to be addressed. Finally, this activity allows teachers to hear about how others integrated CT into science and get ideas to learn from each other.

Human Robot Activity

Summary:
Teachers work in small groups to write programs and direct a fellow teacher (robot) around a course.

Time: 45 - 60 minutes

Instructions:
1. Before the activity, set up the course according to the Robot Course
   a. Note: If available, this activity works best somewhere with visible tiles that can be used as a metric of measurement for movement
2. Assign teachers to or have teachers form small groups of 3-4 teachers and hand out the Group Instructions and Challenges sheet.
3. Teachers select one group member to be the first "robot." The robot teacher moves away from the group to the beginning of the course while the other teachers write the program.
4. The remaining teachers write a set of instructions, or a program, for the robot teacher that will guide the robot teacher through the challenge. The robot cannot turn, it can only move forward, backward, left, or right based on the written program. The program is given to the robot as a set of arrows paperclipped together in the correct directions. Each arrow represents one step of movement.
5. The group gives their robot the instructions beginning with the robot starting on the green arrow.
6. The robot can only move according to the instructions, he/she cannot make any other movements. The robot follows the program one arrow at a time moving in the specified direction. If the robot runs into a wall, obstacle, or doorway, it stops the program with a "fail."
7. Once the robot has successfully completed the course, groups record their program on the challenge sheet.
8. The group selects a new robot for the next challenge and repeats steps 3-7 for all challenges.

Resources:
- Human Robot Group Instructions and Challenges
- Robot Course
- Robot Arrows

Activity Rationale:
The purpose of this activity is to understand programming both from the perspective of the programmer and from that of the computer. Since the participant is acting out the part of the robot, they are able to understand that the computer (or robot) which is running the program has no interpretive power except what is present in the program. The computer (or robot) can only take the steps in the precise order of the program without questioning those steps. This knowledge helps the programmer to understand that all aspects of the program must be considered. This is important because programming is quite a detail oriented activity and it requires planning. To get the robot to go through the maze, the programmer must create a plan (or algorithm), translate it to steps the robot will be able to understand (programming) and then give that program to the robot (program execution). Then, if the execution does not go as planned, going back to the planning step to create a new algorithm, translate that algorithm to a program and rerun the program is the testing life cycle so important in learning to program. Since many plans will not be successful on the first try, this lesson also teaches the persistence that is needed to repeat this program testing cycle multiple times until the robot is successfully able to get through the maze. The increased level of difficulties of the challenges allows the learners to build their skills at making more elaborate plans for their robot.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Advice for Future Use:
This activity demonstrates algorithms, an essential component of computational thinking. But, the activity does not directly tie into science learning. Following the activity, have a conversation with teachers about how this activity could be used with their students and how the teachers would adapt the activity to science contexts or make direct connections with science learning.

Jigsaw Challenge: *Synthesize a Story*

Summary:
Teachers work in groups to synthesize a story that has been decomposed into parts. Collaborating as a team, teachers read through snippets of an article about DNA sequencing and work to put the story together in the correct order. After reconstructing the story, the teachers reflect with facilitators about the CT challenge.

Time: 20 minutes

Instructions:

1. Break teachers into teams or have them divide into teams. Ideally groups will be around 4 people.
2. Introduce the challenge to teachers explaining that they are going to synthesize a story and put it back into the correct order like a jigsaw puzzle.
3. Distribute the article already decomposed into pieces.
4. Teachers work together to realign the story into the correct order.
5. After the teachers have finished putting the story back together, lead teachers in a reflection about the CT challenge.
    a. How were you using CT concepts/capabilities as you engaged in the challenge? (Write on the board any concepts of CT brought up and used during the activity and have a facilitator arrange terms together to create idea clusters.)
        i. This activity models the use of problem decomposition, troubleshooting, collaboration, and developing code in parallel (using parallel processing).
    b. What challenges did you encounter and how did you address them?

Resources:

- Decomposition Article

Activity Rationale:
This activity gets the teachers thinking about different ways the same process can be done. Some groups approached it very linearly, while others worked to make larger chunks, and still others came up with a system closer to parallel processing working on their chunks concurrently. This idea of thinking about how the process is done and what features or which processes can be implemented by a computer help the teachers see the distinction of how CT is already being applied in some activities and that thinking about the process of how something is done is more important to CT integration than the actual completion of the task.

Advice for Future Use:
This activity demonstrates problem decomposition, an essential component of computational thinking. But, the activity does not directly tie into science learning. Following the activity, have a conversation with teachers about how this activity could be used with their students and how the teachers would adapt the activity to science contexts or make direct connections with science learning.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Systems Thinking Challenge Stations

Summary:
Teachers complete a rotation of stations to experience CT skills used in science, specifically focused on systems thinking. The stations focus on CT skills and thinking within CS and as it applies to science. After teachers have completed each station, the group discusses their use of CT practices and how the activities can be adapted to their science classrooms.

Time: 30 minutes (6-7 minutes per station with 1 minute for rotations) + 10 minute Debrief

Instructions:

1. Break teachers into small groups and describe the activity for today: station rotations specifically focused on systems thinking within science contexts. Emphasize that some examples are computer science examples, but they are shown for the purpose of seeing the kinds of skills and thinking in computer science that can be applied to science.
2. Teachers rotate to each of the four stations:
    a. Station 1: Finite State Machines Activity
    b. Station 2: 20 Questions Activity
    c. Station 3: Google Trends Activity
    d. Station 4: Data Compression Activity
3. Teachers work at each station for 6-7 minutes and then rotate to the next station.
4. After the rotations, teachers come back together as a full group to debrief the activity. Make sure to emphasize that these activities were not designed to teach systems thinking, but rather to model systems thinking in science.
5. Have teachers write the CT practices they used on sticky notes and bring them to a broad. Organize the sticky notes by categories.
6. Discuss (small group or full group):
    a. What challenges did you encounter and how did you address them?
    b. How could you adapt these activities for other science concepts or for your students' age group?

Station Instructions:
Station 1: Finite State Machines Activity

1. Explain finite state machines to teachers.
    a. Finite state machines are a computational model that assists people in understanding and learning to apply sequential logic in a very basic way while also addressing the understanding that some things are possible with the restrictions of the model and other things are not. Computers and computer programming is often based on the decisions made based on the current state and the options available at that state.
    b. Display and demonstrate the finite state machine diagram for this station.
2. Distribute the Finite State Machines Handout to Teachers
3. Teachers use the displayed finite state machine diagram to determine whether the given words would be accepted by the finite state machine.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

4. Then, have teachers try to create their own word that would be accepted and one that would not be accepted by the machine.
5. Discuss how this applies to science such as, how decisions can only be made based on the information currently available. Since we have to make decisions based on the data currently available and never have complete knowledge, the decision making process is similar to being in various states of the finite state machine.

Station 2: 20 Questions Activity

Based on the Exploration: Classifying Objects Lesson

1. Teachers are given images of 8 different living organisms and a conditional decision tree.
2. The facilitator identifies one organism to think of, but does not tell the teachers what that organism is.
3. Teachers have to write a series of yes and no questions on their decision tree to determine what organism the facilitator is thinking about. Teachers can work alone or with a partner.
4. Once all of the teachers have written questions, they check whether their questions were accurately able to deduce the correct animal.
5. Explicitly point out the connection to dichotomous keys in science and discuss how they have been used as a tool to identify and classify organisms.

Station 3: Google Trends Activity

1. Explain Google Trends to teachers and demonstrate how it works with a live example.
2. Give each teacher a copy of the four trends graphs and show them a list of the four possible words the trends could represent.
3. Have teachers individually try to identify which trend aligns to each word.
4. By graph, ask teachers to individually share which word they assigned and why. Discuss what data and knowledge they used to determine the answers. Consider how these graphs demonstrate behavior over time. What changed in each graph to cause the spikes? Why did it change? Why is that change important?
5. Discuss how this could be used within science classrooms when discussing natural disasters, data analysis, and climate change amongst other topics.

Station 4: Data Compression Activity

Based off a lesson from Google Education Data Compression Lesson

1. Handout the Data Compression Activity Handout
2. Talk through the ideas of compression and masking when sending and storing information. Compression and masking are necessary because of data storage and transmission limitations, but often cause differences in images that are undetectable to the human eye. Knowing what can be masked and what can not be masked to limit distortion is the goal of the testing in this exercise.
3. Have a brief explanation about the fact that pictures are actually a sequence of pixels which each contain features like color and intensity.
4. Have a brief discussion about ways that data might be reduced in size and what impacts those options for reduction might have.
5. With small groups of teachers working together, have them complete the handout using the on-line application: https://example.pencilcode.net/edit/data-compression where they can change line 26 to

the different values given to see how the filtered photograph is different from the original.
6. Discuss the implications including when those losses would be acceptable and when they would not be.

Resources:

- Station #1: Finite State Machine Diagram
- Station #1: Finite State Machine Handouts
- Station #2: Animal Categorization Handout
- Station #3: Trend Graphs
- Station #4: Data Compression Activity Handout

Activity Rationale:
These activities give the participants basic insights into various aspects of how computers need to do things differently than humans do. The idea that the decision space is limited based on the model, the planning of the sequential questions, the classification and analysis of large data sets and the limitations of storage leading to compression are all examples of this. Having teachers experience these limitations and aspects of computing on this small scale rather than just explaining each helps the teachers to see how students need to understand and internalize the concepts in order to be able to apply them in their own lives.

What's My Mystery STEM Career?

Summary:
Teachers are assigned a STEM career that uses computational thinking, but this career is kept a mystery from them. The teachers must guess their mystery STEM career by asking peers yes or no questions about the career and its use of computational thinking. Once they have guessed their mystery career, teachers work with others sharing their career to determine CT practices used within the career.

Time: 15 minutes

Instructions:

1. Explain the activity to teachers. Describe that teachers will be assigned a mystery career and must try to figure out their career. Explicitly state that each of these STEM careers requires computational thinking practices.
2. A facilitator places a sticky note on the back of every teacher with the name of one of the STEM careers. This is the teachers' mystery career.
3. Teachers move around the room and ask each other yes or no questions to determine their mystery career. The questions should be about the job tasks of the career and the ways that the mystery career uses computational thinking.
4. Once a teacher has identified their mystery career, they must find the other teachers in the room that share that mystery career.
5. When all teachers who share a mystery career are together, they work together to identify ways other than those listed on the handout that the job uses computational thinking practices within normal job activities using the CT Framework.
6. Pass out the Mystery STEM Career Answer Key to teachers and invite them to add additional practices that are not on the handout.

Resources:

- Mystery STEM Career Answer Key
- CT Framework

Activity Rationale:
Computing careers and the integration of CT within careers are much broader than people first think and are growing every day. This breadth of computing careers is an important concept to get to elementary school children so they are thinking about themselves as possible computing professionals or professionals who use computing in their careers, even if they also have other interests. This broad definition of computing careers also has potential to improve the diversity of people who are pursuing computing and tech careers. This activity is a fun game to allow teachers to think about how CT concepts and skills can be applied to these other types of computing jobs and therefore broaden their minds about how computing is applied throughout our society.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Full Name: ___________________________

Please read the following vignette.

Ms. Smith planned a lesson for her 3rd grade students to learn about living beings. The first part entailed going outside and observing plants and animals such as squirrels and birds. Students wrote their observations on a notepad, where they had two tables: one for "living" and one for "non-living". Ms. Smith asked students to look at some rocks with moss and ask them if either of them was alive. Some students said yes, and others said no.

When they returned to the classroom, Ms. Smith asked the students who believed the moss was alive to show their thinking. Then, she asked students to look at their "living" column and make a list of all things that living beings have in common. Then, Ms. Smith and the students came up with a set of questions that had to be answered one by one to determine if something was alive. For example, they asked: Does it need food? Does it grow? Does it reproduce? The class tried out a few examples (a rock, a bee, and an apple) and only put things in the "living" category if all questions were answered with "yes". Then, each student drew a picture of their favorite living thing and nonliving thing, explaining why each of them was alive or not.

1. In your view, did the students engage in computational thinking in this lesson?

◯ Yes — Answer Questions 2 & 3 below

◯ Maybe — Answer Questions 2 & 3 below

◯ No — Flip Page, Answer questions 4 & 5

2. In which part(s) of the vignette did the students engage in computational thinking?

____

____

____

____

____

3. Which computational thinking practice(s) did the students engage in?

❑ Finding patterns and relationships in data sets	❑ Coding
❑ Sorting data	❑ Test, Adjust to Improve, Retest, Readjust to Improve…
❑ Creating graphs or charts	❑ Identifying quantifiable parts of a system
❑ Using computational simulations	❑ Considering numerical relationships within a system
❑ Assessing computational simulations	❑ Considering how changes to the quantifiable parts contribute to results of the system
❑ Breaking down problems into smaller parts	
❑ Creating step by step instructions to solve a problem	

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

4. How would you modify the lesson to include computational thinking? (Flip the page if you need to review it)

____

____

____

____

____

____

____

____

5. Which computational thinking practice(s) **did you add** to the lesson?

❏ Finding patterns and relationships in data sets ❏ Sorting data ❏ Creating graphs or charts ❏ Using computational simulations ❏ Assessing computational simulations ❏ Breaking down problems into smaller parts ❏ Creating step by step instructions to solve a problem	❏ Coding ❏ Test, Adjust to Improve, Retest, Readjust to Improve… ❏ Identifying quantifiable parts of a system ❏ Considering numerical relationships within a system ❏ Considering how changes to the quantifiable parts contribute to results of the system

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Full Name: ___________________________

Please read the following vignette.

Ms. Nowak designed a lesson for students to understand the relationships between precipitation and temperature. She asked students to use predictions from Weather.com to enter data into an Excel chart (Figure 1).

TIME	DESCRIPTION	TEMP	FEELS	PRECIP	HUMIDITY	WIND
10:00 AM SAT	Cloudy	32°	22°	20%	83%	ESE 16 mph
11:00 AM SAT	Showers	34°	23°	50%	80%	E 16 mph
12:00 PM SAT	Cloudy	33°	22°	25%	81%	ESE 17 mph
1:00 PM SAT	Rain	33°	24°	70%	80%	E 17 mph
2:00 PM SAT	Snow	31°	25°	95%	82%	E 16 mph
3:00 PM SAT	Rain	32°	25°	75%	82%	E 17 mph
4:00 PM SAT	Snow	31°	25°	100%	83%	ENE 14 mph

Precip. Type	Temp	Precipitation
None	32	20%
Rain	34	50%
None	33	25%
Rain	33	70%
Snow	31	95%
Rain	32	75%
Snow	31	100%

Ms. Nowak asked the students, *what do you notice about the temperature when the scientists predict rains versus when it snows?* To answer this question, the students used Excel to group the times that rain or snow was predicted. They represented warmer or cooler temperatures with different colors (Figure 2). The students explained how every time snow was predicted, the temperature was below 32 degrees, which is the temperature where water freezes!

Precip. Type	Temp	Precipitation
None	32	20%
None	33	25%
Rain	34	50%
Rain	34	70%
Rain	32	75%
Snow	31	95%
Snow	30	100%

Finally, Ms. Nowak asked the students to consider the probability that it would precipitate. She asked students, *out of all the days when scientists predicted* ***snow or rain****, what was the* ***lowest*** *precipitation probability?* Using data from the spreadsheet, students identified 50% as the lowest value. Ms. Nowak then asked students to predict whether it would rain, snow, or neither given the temperature and precipitation probability by completing the chart below. Students then discussed their predictions.

Predict Precip. Type	Temp (F)	Precipitation Probability
None	25	48%
Rain	34	82%
Snow	30	60%
	35	20%
	32	50%

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

1. In your view, did the students engage in computational thinking in this lesson?

◯ Yes → Answer Questions 2 & 3 below

◯ Maybe → Answer Questions 2 & 3 below

◯ No → Use other Page, Answer Qs 4 & 5

2. In which part(s) of the vignette did the students engage in computational thinking?

____

____

____

____

____

____

____

____

3. Which computational thinking practice(s) did the students engage in?

❑ Finding patterns and relationships in data sets ❑ Sorting data ❑ Creating graphs or charts ❑ Using computational simulations ❑ Assessing computational simulations ❑ Breaking down problems into smaller parts ❑ Creating step by step instructions to solve a problem	❑ Coding ❑ Test, Adjust to Improve, Retest, Readjust to Improve… ❑ Identifying quantifiable parts of a system ❑ Considering numerical relationships within a system ❑ Considering how changes to the quantifiable parts contribute to results of the system

4. How would you modify the lesson to include computational thinking?

____

____

____

____

____

____

____

____

5. Which computational thinking practice(s) **did you add** to the lesson?

❑ Finding patterns and relationships in data sets ❑ Sorting data ❑ Creating graphs or charts ❑ Using computational simulations ❑ Assessing computational simulations ❑ Breaking down problems into smaller parts ❑ Creating step by step instructions to solve a problem	❑ Coding ❑ Test, Adjust to Improve, Retest, Readjust to Improve… ❑ Identifying quantifiable parts of a system ❑ Considering numerical relationships within a system ❑ Considering how changes to the quantifiable parts contribute to results of the system

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Full Name: ___________________________

Please read the following vignette.

Mr. Benjamin is teaching his 2nd grade class about pollination. He wants his students to understand that some plants use pollinators, such as bees, to reproduce. First, Mr. Benjamin shows his students a video where a scientist shows footage of bees entering multiple flowers while she explains the process of pollination.

Then, Mr. Benjamin asks the students, “how do you think pollen sticks to the bee’s body?”. To explore possible explanations, the students design “pollen collectors.” Students are provided several materials: carpet, sponge, plastic, paper (to represent the bee’s body) and a container of wet sand (to represent the flower/pollen). Each group tests how much “pollen” each material collects. They discuss results as a class. Mr. Benjamin then explains the similarities between these materials, the bodies of bees, and the stickiness of pollen.

The next day, Mr. Benjamin asks the students to engage in a simulation activity about the process of pollination. In the activity, each student is assigned a role as a flower, pollen or bee. When a child playing a bee visits a child representing a flower, another child representing pollen will now follow the “bee” anywhere it goes until it visits another flower. The goal of each group of “bees” is to get each “pollen” to as many different “flowers” as they can. Finally, Mr. Benjamin explains how the activity is like the way bees behave in the world—the bees are not *trying* to pollinate trees; they just happen to do so while they feed on the flowers.

---

1. In your view, did the students engage in computational thinking in this lesson?

◯ Yes	◯ Maybe	◯ No
Answer Questions 2 & 3 below	Answer Questions 2 & 3 below	Flip Page, Answer questions 4 & 5

---

2. In which part(s) of the vignette did the students engage in computational thinking?

____

____

____

____

3. Which computational thinking practice(s) did the students engage in?

❑ Finding patterns and relationships in data sets	❑ Coding
❑ Sorting data	❑ Test, Adjust to Improve, Retest, Readjust to Improve…
❑ Creating graphs or charts	❑ Identifying quantifiable parts of a system
❑ Using computational simulations	❑ Considering numerical relationships within a system
❑ Assessing computational simulations	❑ Considering how changes to the quantifiable parts contribute to results of the system
❑ Breaking down problems into smaller parts	
❑ Creating step by step instructions to solve a problem	

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIGCT: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

4. How would you modify the lesson to include computational thinking? (Flip the page if you need to review it)

____

____

____

____

____

____

____

____

5. Which computational thinking practice(s) **did you add** to the lesson?

❑ Finding patterns and relationships in data sets ❑ Sorting data ❑ Creating graphs or charts ❑ Using computational simulations ❑ Assessing computational simulations ❑ Breaking down problems into smaller parts ❑ Creating step by step instructions to solve a problem	❑ Coding ❑ Test, Adjust to Improve, Retest, Readjust to Improve… ❑ Identifying quantifiable parts of a system ❑ Considering numerical relationships within a system ❑ Considering how changes to the quantifiable parts contribute to results of the system

Full Name: ___________________________

Please read the following vignette.

Ms. Spring is teaching her 5th grade class about energy. Her goal is for students to understand that energy in animals' food was once energy from the sun. Ms. Spring's students generally know that plants, through photosynthesis, create glucose (sugar), which is used by cells. They also know that some animals eat plants, and that other animals eat smaller prey. But, she wants students to understand the big picture; the connection from the sun to the carnivore.

To reinforce the idea that energy in animals' food was once energy from the sun, Ms. Spring has students use Scratch to create a story that shows how energy is converted in each step. The story must start with the sun, have 1 or 2 energy conversions, and end in an animal eating another animal. The students create their stories by programming animations that represent the conversion of energy.

Ms. Spring uses the stories to assess the students' understanding. For example, she notices that Timmy created a story where the sun heated up a fish that a bird was eating, and Timmy said that it was that heat energy that is absorbed by the animal. Ms. Spring uses this opportunity to talk to the class about the differences between energy from heat and energy from food.

---

1. In your view, did the students engage in computational thinking in this lesson?

◯ Yes	◯ Maybe	◯ No
⬇	⬇	➡
Answer Questions 2 & 3 below	Answer Questions 2 & 3 below	Flip Page, Answer questions 4 & 5

---

2. In which part(s) of the vignette did the students engage in computational thinking?

____

____

____

____

____

3. Which computational thinking practice(s) did the students engage in?

❑ Finding patterns and relationships in data sets ❑ Sorting data ❑ Creating graphs or charts ❑ Using computational simulations ❑ Assessing computational simulations ❑ Breaking down problems into smaller parts ❑ Creating step by step instructions to solve a problem	❑ Coding ❑ Test, Adjust to Improve, Retest, Readjust to Improve… ❑ Identifying quantifiable parts of a system ❑ Considering numerical relationships within a system ❑ Considering how changes to the quantifiable parts contribute to results of the system

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

4. How would you modify the lesson to include computational thinking? (Flip the page if you need to review it)

____

____

____

____

____

____

____

____

5. Which computational thinking practice(s) **did you add** to the lesson?

❑ Finding patterns and relationships in data sets ❑ Sorting data ❑ Creating graphs or charts ❑ Using computational simulations ❑ Assessing computational simulations ❑ Breaking down problems into smaller parts ❑ Creating step by step instructions to solve a problem	❑ Coding ❑ Test, Adjust to Improve, Retest, Readjust to Improve… ❑ Identifying quantifiable parts of a system ❑ Considering numerical relationships within a system ❑ Considering how changes to the quantifiable parts contribute to results of the system

Your Name: ___________________________________

Group Members: ____

Part 1: Computational Thinking-Infused Elementary Science Lesson Seed

Directions: Complete the lesson seed by answering questions #1-3 below during the workshop in your small groups. You will complete a lesson seed during each workshop. One time this semester, you are expected to build on the lesson seed to develop a lesson plan that you teach to your class. For this lesson, you should also complete Part 2.

1) Describe the objective(s) of the lesson.

2) Describe the instructional procedures for your computational thinking-infused elementary science lesson (use back for more space).

3) Identify the computational thinking practice(s) in the lesson?

Using data

- ❑ Finding patterns and relationships in datasets
- ❑ Sorting data
- ❑ Creating graphs or charts

Computational Simulations

- ❑ Using computational simulations
- ❑ Assessing computational simulations

Programming

- ❑ Breaking down problems into smaller parts
- ❑ Creating step-by-step instructions to solve a problem
- ❑ Coding
- ❑ Test ⟶ Adjust to improve ⟶ Retest ⟶ Readjust to improve

Computational Systems Thinking

- ❑ Identifying quantifiable parts of a system
- ❑ Considering numerical relationships within a system
- ❑ Considering how changes to the quantifiable parts contribute to results of the system

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

This material is based upon work supported by the National Science Foundation under Grant No. 1639891.

Your Name: ___________________________________

Grade: ____________

Part 2: Computational Thinking-Infused Elementary Science Lesson Plan & Reflection

Directions: ONE TIME THIS SEMESTER, you should build on Part 1 to develop a lesson plan that you teach to your class. For this lesson, you should complete questions #1-5 below.

1) *Describe, in as much detail as a carefully designed lesson plan, the instructional procedures for the computational thinking-infused elementary science lesson you completed in your classroom (Note: It is okay to be modified from your lesson seed in Part 1).*

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

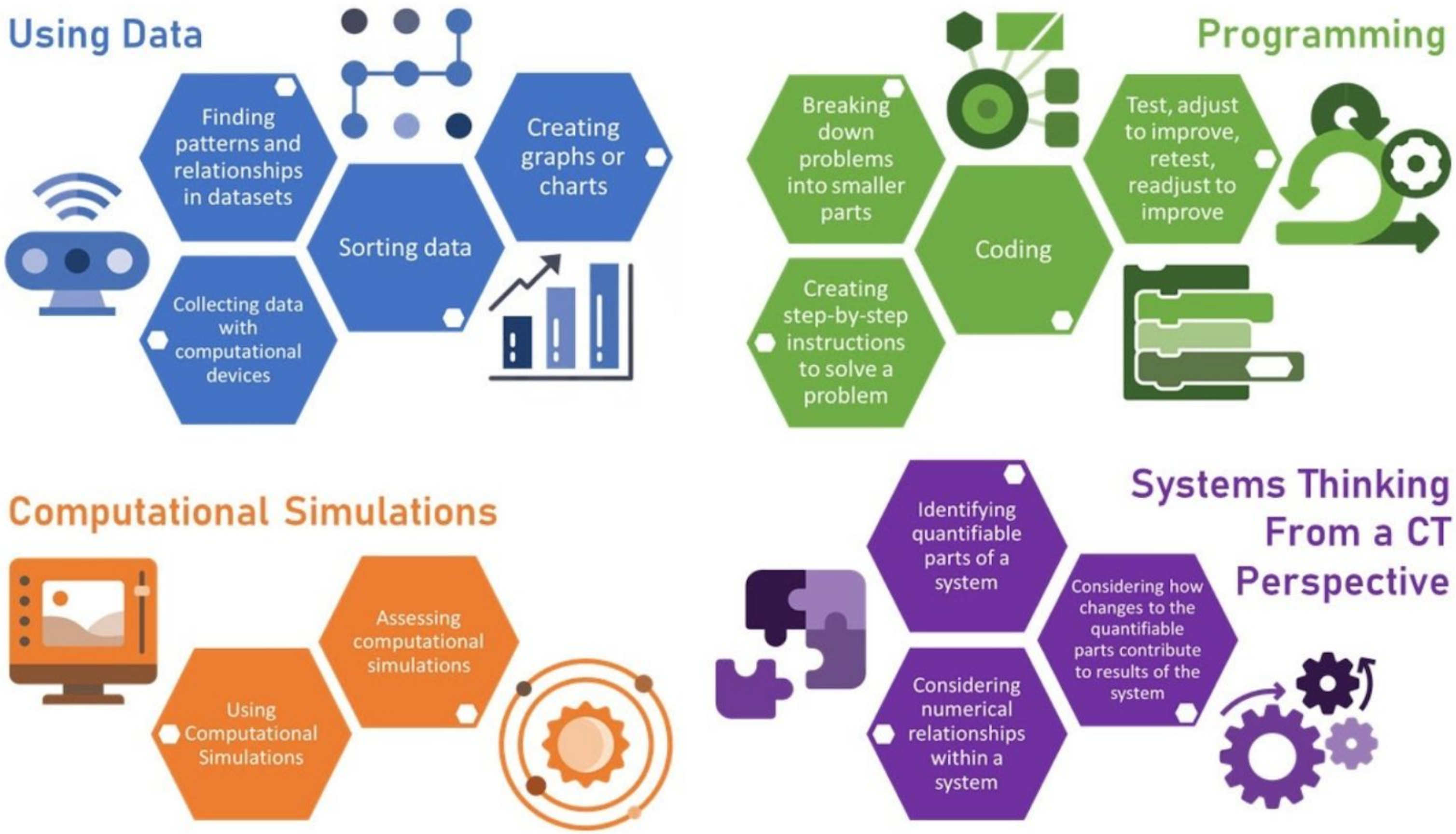
This material is based upon work supported by the National Science Foundation under Grant No. 1639891.

2) *What did your students learn (or not learn) about science? How do you know (what did the students say and/or do)? (Note: Be specific. For instance, you may write down anonymous student quotes, transcribe anonymous student work or describe specific actions of anonymous students during class).*

3) *What did your students learn (or not learn) about CT? How do you know (what did the students say and/or do)? (Note: Be specific. For instance, you may write down anonymous student quotes, transcribe anonymous student work or describe specific actions of anonymous students during class).*

4) *Did you see evidence that some students in your class were more engaged or successful than others? If so, please explain. (Note: Be specific. For instance, you may write down anonymous student quotes, transcribe anonymous student work or describe specific actions of anonymous students during class).*

5) *What would you do the same and what would you do differently next time for this lesson? Why?*



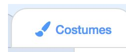
Framework Citation: Ketelhut, D. J., Cabrera, L., McGinnis, R. J., Plane, J., Coenraad, M., Killen, H., & Mills, K. M. (2019). Exploring the Integration of computational Thinking into Preservice Elementary Science Teacher Education. *National Science Foundation STEM+C PI Meeting*. http://stemcsummit.edc.org/slides/DianeJass.pdf

Creating Simulations in Scratch

Today, we will create a simulation where it snows when a hot and cold front touch.

1. Create two sprites representing the warm and cold fronts

Hints:

- You can add a cloud sprite by clicking on the 
- You can edit a sprite by clicking on it and the 

tab
- You can use the text to write "Warm" on the warm front and "Cold" on the cold front.
    - Bonus: In the costumes tab, you can also change the color of the sprite here, edit the shape, and add more to it. You could even draw the symbols for warm and cold fronts if you want!

2. Add a snowflake sprite to represent the snow!

Hints:

- You can find a snowflake in the same place you found the clouds!
    - Did you know: You can search through the available sprites? There is a search bar in the sprite screen.
- If you want, you can edit the snowflake sprite in the 

tab to make it smaller and add copies of the same image to create a sprite with lots of snowflakes rather than just one! Just copy and paste within the costume.

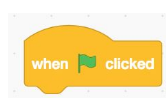
3. Add a backdrop

Hints:

- You can add backdrops by clicking on the 

4. Set up the fronts in locations where you want them to begin and add blocks to Initialize their locations. The fronts should start not touching.

Hints:

- You can drag the sprites around the stage to move them to where you want them.
- Initialize the blocks by telling them where to go to 

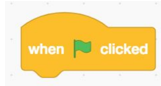
    - Each sprite will need a script telling it where to go when you click the green flag.

***5.* Initialize the location and visibility of the snowflake**

Hints:

- Drag the snowflake around to the location where you want them to appear when the fronts touch. Keep in mind where the fronts will touch to position the snow.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

- Make the snowflake hide 

    - Check the Looks blocks to find the right block to use.

6. Code one front to move to the other front

Hints:

- Look in the Motion blocks to find a block that will make the front move across the screen. There are multiple blocks that you can use!
- Make sure to use an Events block to trigger the front to move. This event could be pressing the spacebar, clicking on the sprite, or clicking any key!

7. When the fronts touch, make the snowflake appear

Hints:

- You can do this in multiple ways! Your solution should include:
    - A condition (if/then or repeat until) from the Control blocks
        - Blocks can be placed inside the c-shaped conditions!
    - A block to sense whether the two fronts are touching from the Sensing blocks
    - A broadcasted message from the Events blocks
    - The snowflake receiving a message (also from the Events blocks) and showing (try the Looks blocks where hide was)

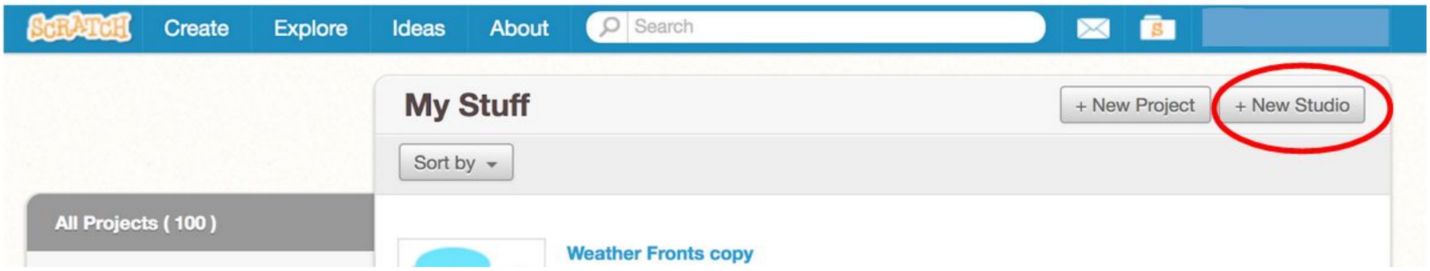
Challenges:

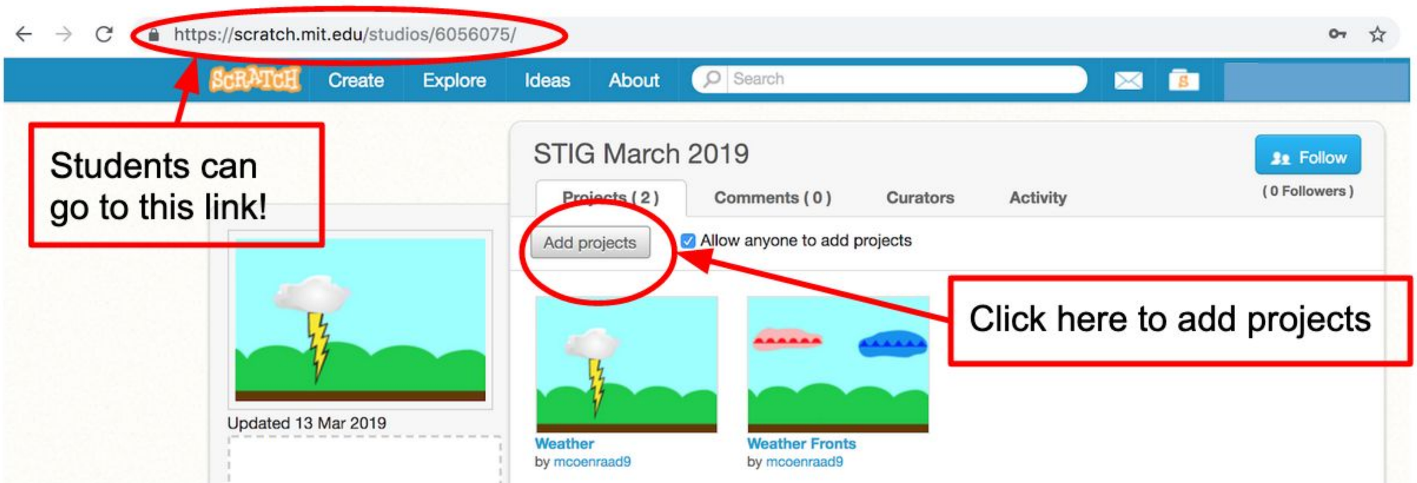
- Change the pace that the front moves across the screen to make it move slower
- Make both of the fronts move toward each other at once
- Angle (try blocks from the Motion category) the warm front when it gets to the cold front and have it go above the cold front.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

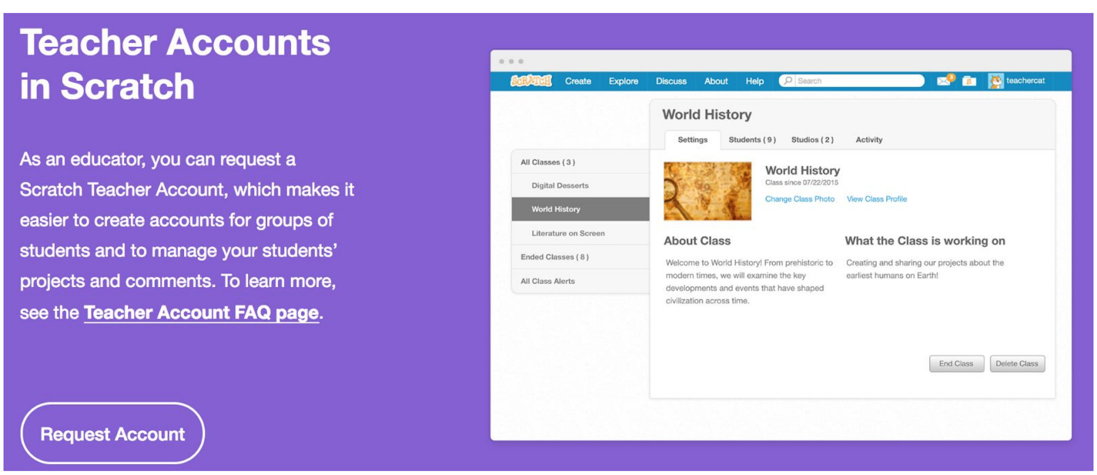
Hints for Using Scratch in the Classroom

You can create a studio in the "My Stuff" page where students can add projects to. This makes it easy to see the work that all of your students have done!



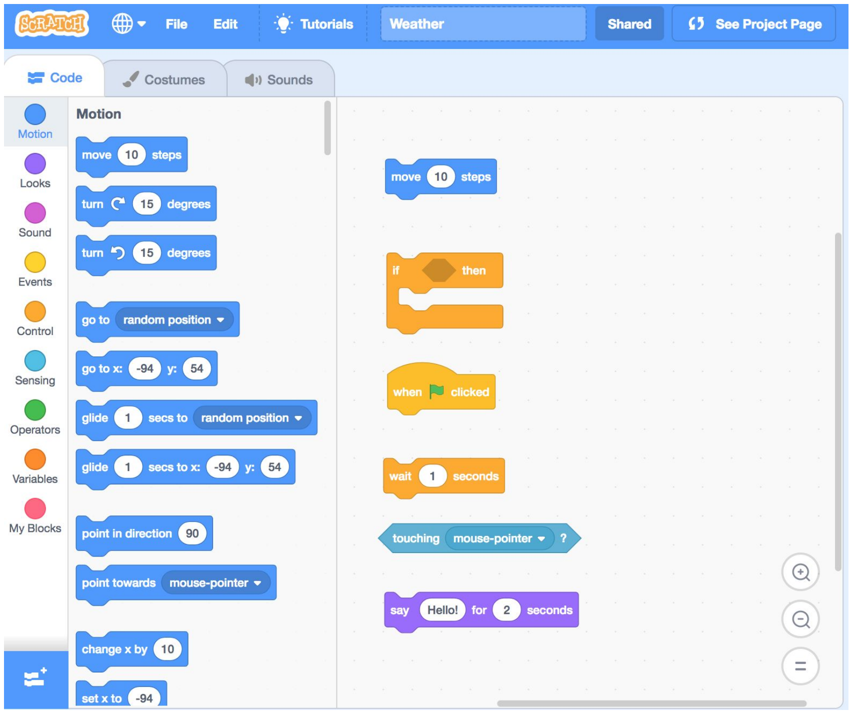


Scratch provides educator accounts to teachers that allow them to manage student accounts and allow easier access to student work. For more information, and other teacher resources, go to: https://scratch.mit.edu/educators



Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

If you want to limit the number of blocks that students have to think about, consider creating a starter project that the students remix with the blocks that they will need on the stage but not put together. Then, students can figure out how to build the code without having to search through all of the blocks.



To introduce students to blocks, have them explore an already made project by making some changes and seeing how it affects the program. Like this, but don't give students the new code, they have to make it!:

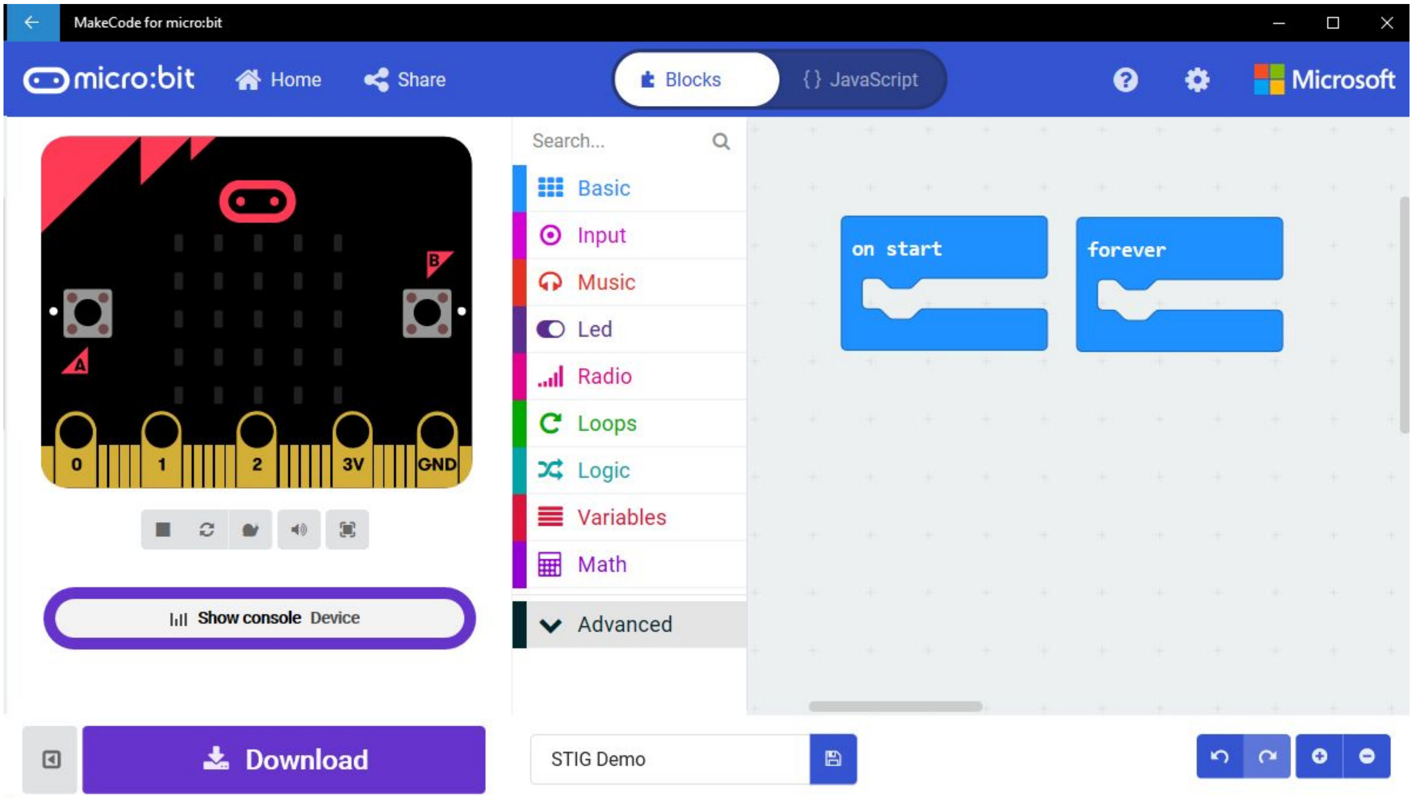
Students start with this:	Have students change the repeat block to repeat 30 times like this:	Now, change the repeat block back to 10 times and change it to move 30 steps. What happens?
		

Leave time for students to explore and play around. Scratch is great for inquiry learning!

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Measuring Light and Temperature with the micro:bit

We're going to use MakeCode, which is similar to Scratch, but is made for the micro:bit.



While MakeCode has a cool simulator, today we want to collect *real* data. We want to do things in the real world! So, whenever we create a program, we're going to Download the program to the micro:bit. Downloading the program is like teaching it to the micro:bit. It won't know what to do until you teach it your program!

The micro:bit can sense temperature and light, but how can we use it to collect that data so we can use it later?

We can start by telling the micro:bit to show the light level with the LED lights, then wait 2 seconds… and show it again. Try it out!

How can we test this? What can we change to see different numbers?

How can we capture this data to use it later?

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Let's use another technique to capture the data. Instead of showing the light level on the LEDs, what if we could just send it back to the computer? Try the "serial write line" block. See what else pops up next to the simulator.

Let's tackle some science questions. How could students use the micro:bit to answer:

1. What is the warmest time of the day?
2. What is warmer: sunrise or sunset?
3. Is the time when the sun shines the brightest also the warmest time of the day?
4. How fast does temperature increase from sunrise to noon?

Pick a question and think about the following points:

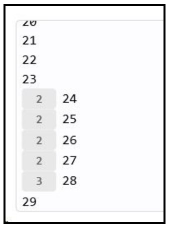
- What do we need to know to answer that question?
- How could we use the micro:bit to capture that data?
- If your question uses light, how can we simulate the sun shining through the day on the micro:bit?

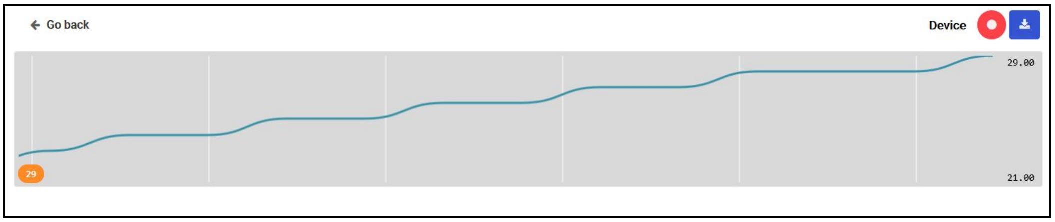
Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Advanced uses of the micro:bit with temperature or light levels.

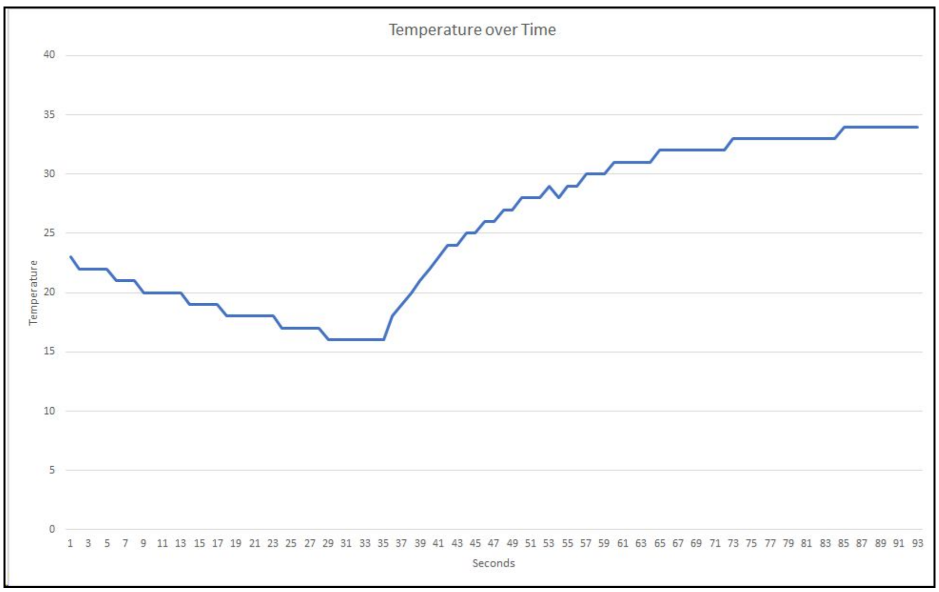
This program will capture the temperature, instead of light levels, every two seconds.

Play around to test its sensitivity: how quickly does it detect changes?





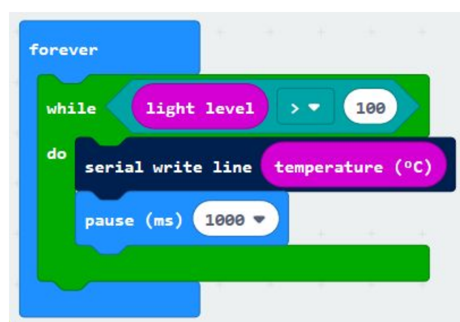
MakeCode can collect and graph the data for you, but you can get even fancier!



This download data button will create an Excel file for you, which you can graph however you want.

You could also create programs where the temperature is collected depending on the light levels. For example, here is a program that only collects the temperature data if the light level is above a certain level (so, if it's during daytime, for instance):

Use a flashlight to increase the light level of the micro:bit and test it out!



Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

WHEN were the weather fronts?

In this activity, we are going to play with some weather data!

Getting Started:

1. Access the WHEN were the Fronts Data
    a. You should see weather data entered to a Google Sheet. This data is the temperatures from the past month, from weather.com in the "Monthly" section.
2. Make a copy of the Google Sheet! (File → Make a copy)

Questions:

1. What were the highest and lowest temperatures?

2. When did temperatures change from high to low?

3. How large were temperature differences when they changed?

Activity 1: Sorting numbers from high to low

1. Highlight all of the data.
2. Click "Data"
3. Click "Sort range"
4. Click "Data has title row"
5. Select "Temperature" from the drop down menu
6. Click "Sort." You should now see the temperatures arranged from hottest day to coldest day.

Activity 2: Visualizing data with color

This time, let's visualize the different temperatures with color. Click "undo." You should now see the temperature arranged again by date.

1. Highlight the temperature column.
2. Click "Format"
3. Click "Conditional formatting"
4. Click "Color scale."
5. Click Done. You should now see the coldest temperatures in a different color than the warmest temperatures.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Activity 3: Finding differences between cell values

Now, let's compute the day to day difference in temperatures. We are going to do this by entering the change in temperature in the last column, "Difference in temperature from the previous day." To do this,

1. Click cell D3. Type "=" (equals sign)
2. Then Click cell C3.
3. Type "-" (subtraction sign)
4. Then Click cell C2.
5. Press enter. You now see the difference between the temperature on February 13 and 14. We are now going to copy this equation to see the difference for all of the dates.
6. Click cell D3
7. Click and drag with the + cursor that appears when you hover over the bottom right corner from D4 to D30. You should now see the difference for every day of the past month.

Name: ________________________________

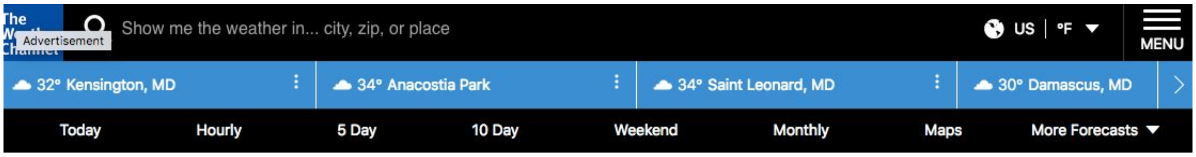
WHEN were the weather fronts?
Reflection

In this activity, did CT allow you to do something with science that you may not have been able to otherwise? Explain.

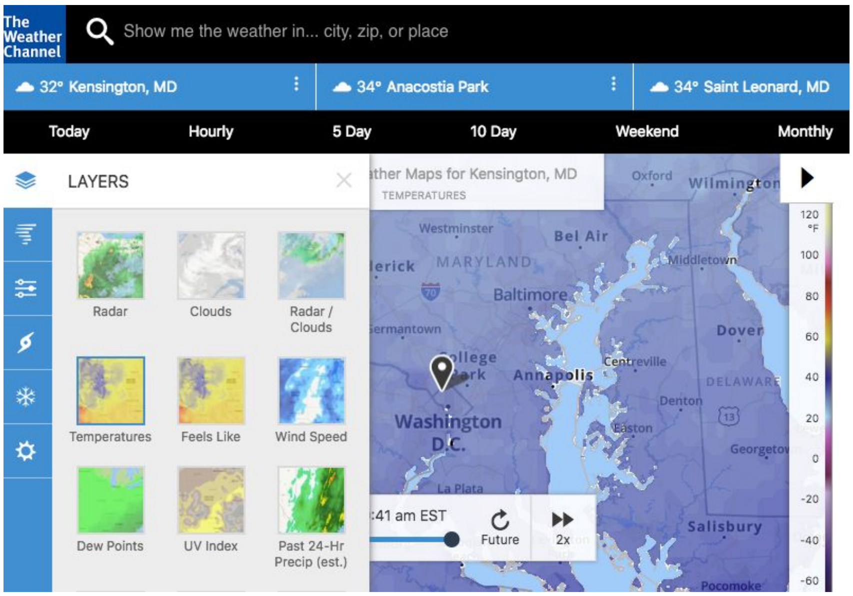
Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

WHERE are the weather fronts?

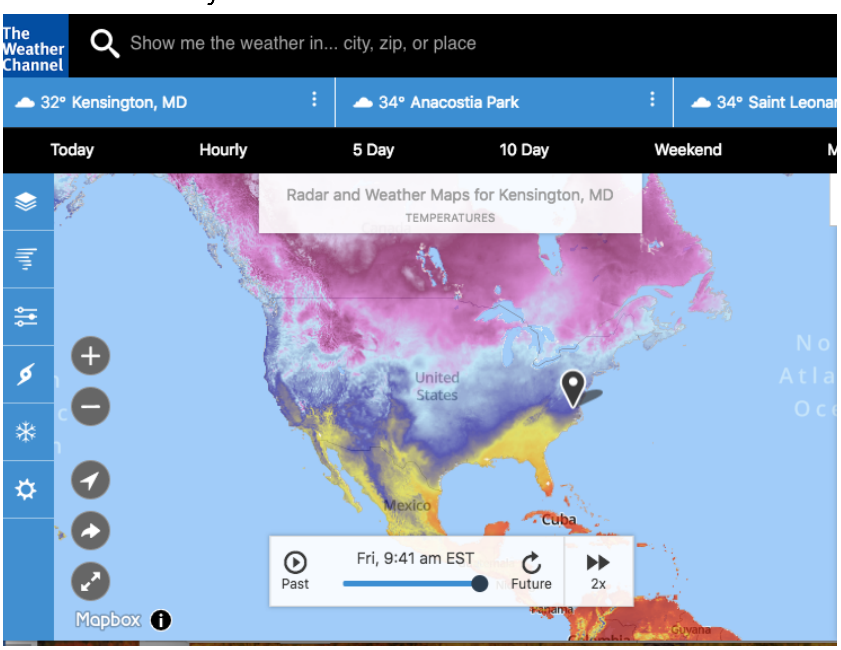
1. Go to https://weather.com/
2. Click "Maps"



3. Click "Layer" on the left hand navigation pane. Scroll down to select "Temperatures."



4. Zoom out so you can see the entire continental United States.



Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

5. Based on the temperature data you see in real time on weather.com (not in the picture above), draw where you predict weather events (rain, snow, sleet) may be happening in the United States
HINT: think about where the weather fronts are.

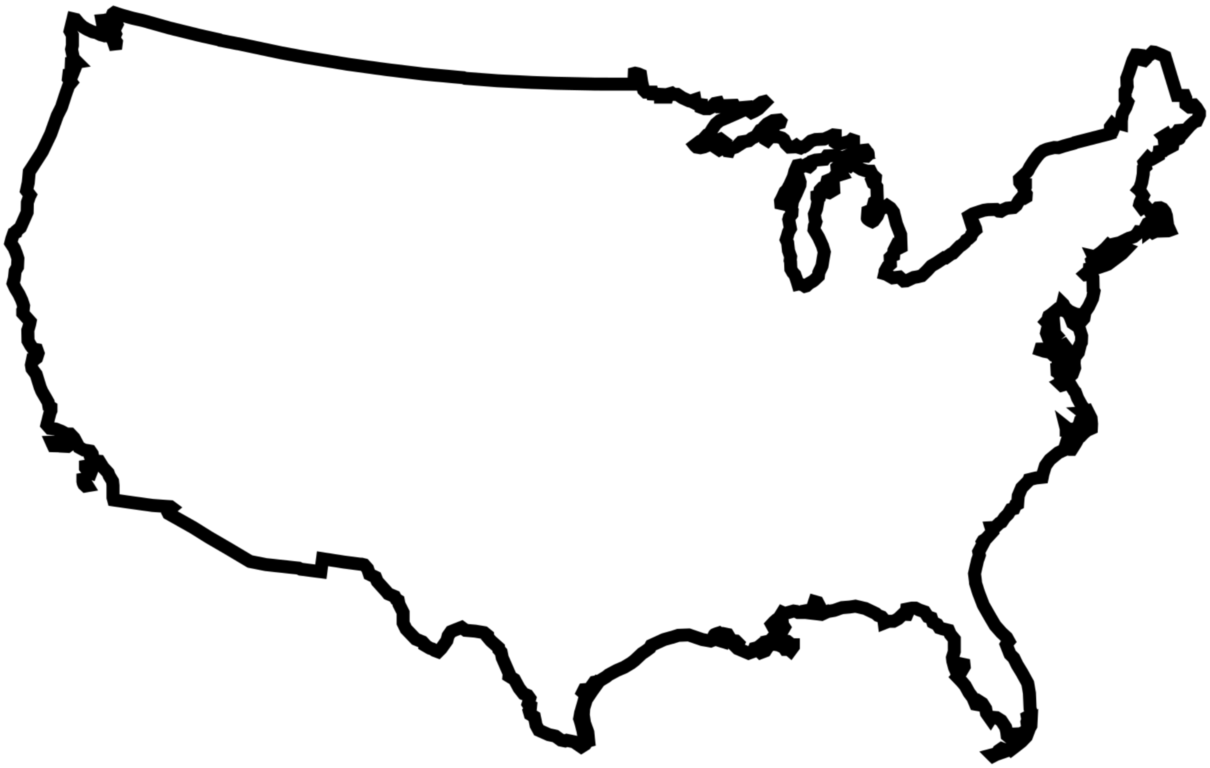
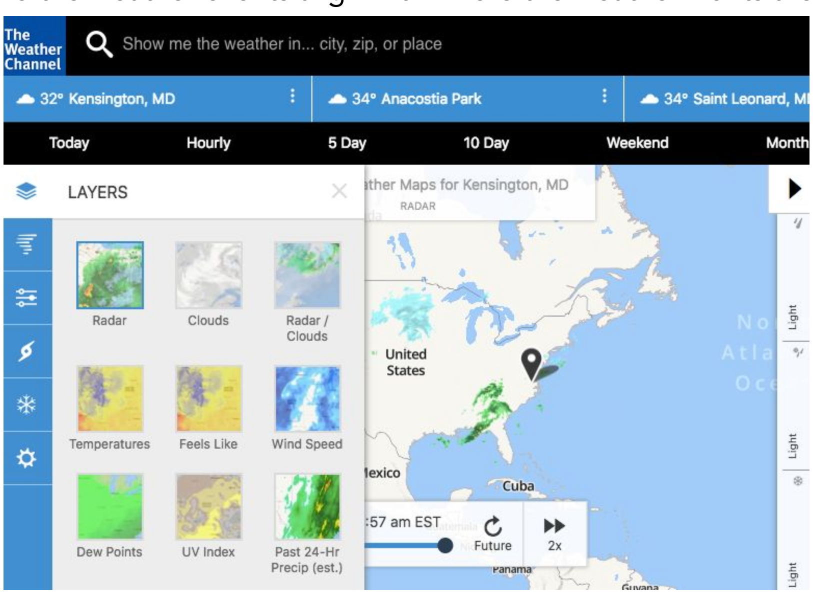


Image from: https://commons.wikimedia.org/wiki/File:US_mark.svg
User: Lokal_Profil / CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)

6. Check your prediction! Go back to "Layers" and this time select "Radar."
Do the weather events align with where the weather fronts are occurring?



Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Name: ________________________________

WHERE are the weather fronts?
Reflection

In this activity, did CT allow you to do something with science that you may not have been able to otherwise? Explain.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Exploring Computational Models and Simulations

Part 1:

Explore the simulations:

- EarthViewer
- Gravity & Orbits

With your group, discuss:

- How do these simulations allow students to explore phenomena that they are unable to observe?
- How could a simulation such as this be used with your students?
- If and how do these simulations provide opportunities for children to engage in science learning?
- If and how do these simulations provide opportunities for children to engage in computational thinking?

Part 2:

Go to: http://new.learningscience.org/wp/
Explore the website for simulations for your grade and/or content.

With your group, discuss:

- If and how does each simulation provide opportunities for student learning?
- If and how are the simulations you are finding aligned to our criteria of a simulation (variables that can be manipulated, dynamic changes represented based on selected variables)?

Part 3:

Complete the written reflection (use the back of this paper, if necessary).

Which simulation(s) did you explore? Do they provide opportunities for children to engage in science learning that they could not with a non-computational tool (such as a physical model)? Explain.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Your Name: ___________________________________

Grade level of lesson: __________________________

Topic: _______________________________________

Computational Thinking Lesson Plan
Feedback & Reflection

You have worked hard this semester to develop ideas to integrate computational thinking into your science teaching! Now that you have bravely integrated one of these ideas in your classroom, let's reflect in order to learn from and build on the experience.

Complete the following questions individually, before sharing your lesson with your fellow CT educators in small groups:

1. What were the strengths of your lesson?

2. What components of your lesson may need more attention and ideas/suggestions?

3. How confident are you that your students engaged in computational thinking? Explain.
   HINT: Use the CT Framework handout for reference.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

This material is based upon work supported by the National Science Foundation under Grant No. 1639891.

Procedure

1. Share your lesson with your group using the following questions as a guide (around 5 minutes).
    - Tell us about your lesson
        - What did you do?
        - What lesson seed did you use? Did you adapt? If so, how?
        - What CT practice(s) did you include?
    - How did it go?
        - What worked? What failed?
        - For whom?
        - How do you know?
2. Your fellow CT educators will offer reflections, as appropriate, on the lesson. Phrase comments in the form of praise and leading questions.
3. Lead a conversation with your group about how they see the integration of CT in the science lesson and how it might be improved. *HINT: Use CT Framework handout for reference.*
4. Complete the written reflection below to summarize your conversation.

Praise (What went well and why)

Example: *Your use of the public data chart was very effective. Students felt some real ownership in the data they created and wanted to find some rule to organize the information.*

Suggestions (Ideas to consider for next time)

Examples: *Do you think it would have helped if …? What do you think would have happened if …?*

Integration of CT (how was CT already there and how might it be improved?)

Examples: *This lesson illustrated computational thinking because students were manipulating a computational simulation to understand how the planets move around the sun. The CT could have been improved if students were considering how mathematical relationships were used to create the simulation.*

DNA Sequencing Article for Problem Decomposition Activity

Article text from: https://arstechnica.com/science/2009/09/a-brief-guide-to-dna-sequencing/

Set 1
and the human genome, whether it's from healthy individuals or cancer cells, has received special attention. A dozen or more companies are attempting to bring new but all the complexity is generally there to just sequence lots of molecules in parallel; the actual process remains pretty simple. In a series of articles, we'll start it's one of the hottest high-tech fields on the planet. But, although these methods can differ, sometimes radically, in how they obtain the were used to complete the human genome. From there, we'll spend time on the current crop of "next-generation" sequencing hardware, before going on to simple, the process of sequencing it is straightforward enough that anyone with a basic understanding of biology can probably In a series of articles, we'll start with the very basics of DNA sequencing, and build our way up to the techniques that were used to to bring new sequencing technology to market that could eventually drop the cost of sequencing down to the neighborhood of a new laptop. Arguably, it's one of It's rare for a month to go by without some aspect of DNA sequencing making the headlines. Species after species has seen its genome completed, and the human a long chain of alternating sugars and phosphates, with each sugar linked to one of four bases. Because the chemistry of DNA is so simple, the process years. Anyone who's made it through biology knows a bit about the structure of the double helix. Half of one is shown above, to illustrate its three components: its backbone is made up they obtain the sequence of DNA, they're all fundamentally constrained by the chemistry of DNA itself, which is remarkably simple: a long chain biology can probably understand the fundamentals. The new sequencing hardware may be very complex, but all the hardware, before going on to examine some of the more exotic things that may be coming down the pipeline within the next few years.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Set 2
although they could be potentially be guanine (G), cytosine (C), or thymine (T). In the double helix, the bases undergo base pairing to is base paired with a partial complement of A’s on an opposing strand. The DNA polymerase, is able to add additional nucleotides (a sugar + base combination) under two ready to react, allowing the cycle to continue. By moving down the strand and repeating this reaction, a new molecule of DNA with a specific that make copies of DNA within a cell are so efficient that biologists have used a modified polymerase to perform sequencing. A string of T’s is base paired each of the two halves as a template for an new opposing strand; the base pairing rules ensure that the copying is exact, except is made up of alternating sugars (blue) and phosphates (red), and each sugar is linked to one of four bases (green). In this case, all of the bases shown are adenine (A), although they could with a specific sequence is created. From a sequencing perspective, having a new copy of DNA isn’t especially helpful. What we want to under two conditions: they’re in the "triphosphate" form, with three phosphate groups in a row, and they base pair we want to know is what the order of the bases along the strand is. Sequencing works because we can get the process to stop in they base pair successfully with the complementary strand. The polymerase causes the hydroxyl group (OH) at the end of the existing strand to react with the react with the triphosphate, linking the two together as part of the growing chain. When that reaction is done, there’s a new hydroxyl group ready to react, base pairing to partners on the opposite strand: A with T, C with G. When a cell divides When a cell divides and DNA needs to be replicated, the double helix is split, and enzymes called polymerases use each of the that the copying is exact, except for rare errors. Historically, DNA sequencing has relied on the exact same process of copying DNA—in fact, the enzymes that make copies

Set 3
a mix of nucleotides where the majority are normal but a small fraction lack the hydroxyl group. Now, most of the time, the
takes place is dictated by the fraction of terminator nucleotides in the reaction mix. Now we just need to know what base is
base can be added to the growing strand normally, but, once in place, the process comes to a crashing halt. We've now stopped the process of
stopped the process of DNA replication. Of course, if
to stop in specific places and identify the base where it stops. The simplest way to do this is to mess with the chemistry. Instead of supplying the DNA with a normal
Of course, if you supply the polymerase with nothing but terminating bases, it will never get very far. So, for a sequencing reaction, researchers use a mix of
the time, the polymerase adds a normal nucleotide, and the reaction continues. But, at a certain probability, a terminator will be put in place, and
given to the polymerase can terminate the reaction. If all the C's, T's and G's are normal, but some fraction of the A's are terminators, then that reaction will produce a
of lengths that slowly tails off as fewer and fewer unterminated molecules are left. The point at which this tailing off takes place is
what base is present when the reaction stops. This is possible by making sure that only one of the four nucleotides given to the
with a normal nucleotide, it's possible to synthesize one without the hydroxyl group that the polymerase uses to add the next base. The base can be
will produce a population of DNA molecules that all end at A. By setting up four reactions, one for each base, it's possible to identify the base at
in place, and the reaction stops. If you perform this reaction with lots of identical DNA molecules, you'll wind up with a distribution of lengths that

Set 4
side of an aqueous polymer mesh (called a gel) and run a current through the solution, the DNA will worm its way through the mesh. you're left with. The negative charge on phosphates makes this easy, since it ensures that DNA molecules will move when back in the early 1990s and, trust me, it was anything but artful. The presence of the DNA, marked by those dark bands, came from a in the same place, otherwise you'll have a collection of molecules with two randomly located ends. This part is easy, since DNA polymerases can only add the base at every position. There are only two more secrets to DNA sequencing. First, you need to make sure every polymerase starts copying in the same The other trick is that you need to figure out how long each DNA molecule is in the large mix of reaction products that you're left with. to store it until it decayed to background. The gels were flexible enough that they would shift or bend at the slightest provocation, making the order of bases difficult to read. But not so can be read off to determine the sequence of the DNA molecule. We're now at the state of the art from when I was a graduate student back in the the mesh. Shorter molecules move faster, longer ones slower, allowing the population of molecules to be separated based on their sizes. with a short DNA molecule that base pairs with a known sequences that's next to the one you want to determine. The other trick can only add nucleotides to an existing strand. So, researchers can "prime" the polymerase by seeding the reaction with a short will move when placed in an electric field. So, if you start the reaction mix on one side of an came from a short-lived radioisotope incorporated into the nucleotides. That meant you had to collect everything involved in the process and pay someone to store it on their sizes. By running the four reactions down neighboring lanes on a gel, you'll get a pattern that looks like the one below, which can be read

Set 5
Go to any outdoor event, and the glow sticks should indicate that it's possible to craft molecules that fluoresce in a variety of different colors.
can put a gate at the end of the gel and image the fluorescent activity there. One by one, based on their size, the different molecules will pass through the gate, and
was this a whole lot more convenient, but it enabled a simple four-fold improvement in throughput. Go to any
The next trick was to get rid of most of the gel. As we noted above, molecules work their way through the gel based on their size, but you needed a
to cut it, and people were beginning to discuss all manners of exotic approaches, like reading single molecules with a scanning-tunneling microscope. Fortunately, a couple of
you could read two hundred bases down each lane, making each gel good for about a kilobase of sequence. The human genome is about 3 Gigabases—clearly, this wasn't going to cut it,
you needed a long gel if you wanted to image a lot of them at once. The solution, it turned out, was not to image them at once—something that, before the
of different colors. By picking four fluorescent molecules that are decently spread out—blue for G's, green for A's, Yellow for T's and red for
of the base there. What once required four separate reactions could now be run at once in a single solution. The next trick was
a couple of changes breathed new life into the old approach. For starters, people got rid of the radioactivity by replacing it with a fluorescent tag. Not only was this a
but not so flexible that they wouldn't tear if suitably disturbed. All told, it took a full day to create something from which, if you were lucky, you could read
that, before the switch from radioactivity to fluorescence, wasn't really possible. All you really need is just enough gel to separate things out slightly. You can put a
and red for C's, for example—and linking them to a specific terminating nucleotide, it's possible to link the termination position with the identity of the base

Set 6
the gate, and glow a specific color based on the base at that position. Instead of a couple hundred bases, it was now possible to get about 700 bases of sequence from a single to have a human genome, but the ability to sequence any human genome, from an individual with a genetic disease to the genome of a cancer cell, in order to of a grad student or technician painstakingly adding everything that was needed into individual tubes, a robot could dispense all the reaction ingredients into a small ready for the same sorts of processes that revolutionized many areas of technology: automation and miniaturization. Instead of a grad in order to personalize medicine. That, once again, has set off a race for new and exotic sequencing technology. came as a computer file, ready to be plugged into various analysis programs. With all of these in place, DNA sequencing was ready for the them into a machine that read out the sequencing information. Large gels were replaced by narrow capillaries. The new of these machines. As the bottlenecks were opened wider, the human genome project shot past its planned schedule, and a flood of from a single reaction. Thanks to digital imaging, the data, an example of which is shown below, was easy to interpret. Sequences came as a a flood of genomes followed. But with the increased progress came increased expectations. Ultimately, researchers didn't just want to have a The new sequencing machines could do all of this for many samples in parallel, and the larger sequencing centers had dozens of these machines. into a small plastic plate that could hold about 100 individual samples. A second robot could then pull the samples and deposit them into a

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Human Robot Group Instructions and Challenges

A couple of rules:

1) For each test run, one member of the group is selected to be the robot, and the robot is only able to follow the directions given to it (it can not make comments or change its programming).
2) For each test run, the robot must start on the square marked with the green arrow and must be facing the direction of the arrow.
3) The goal of each test run is to get to the square marked with red according to the challenge facing the correct way on the green arrow.
4) The robot can only move from square to adjacent square in the direction indicated by the program. (The robot should stop the program with a "fail" if/before it runs into a wall or doorway.)
5) The robot is not able to turn. It can only step one square forward, backwards, left or right as the program indicates that it should do.
6) You will write the program by stacking a group of arrows and clipping them at the top with the paper clip. How the arrow is clipped into the stack tells the direction. Remember all directions are relative to the way the robot is facing.
   a. ⬆ means go forward one square.
   b. ➡ means go right one square.
   c. ⬅ means go left one square.
   d. ⬇ means go back one square.
7) If your program runs where the robot starts on the green arrow (facing the direction of the arrow) and ends where the robot is on the square marked red following the challenge, your program has successfully run. If it ends with the robot on any other square or ends because the robot would bump into a wall or go through a doorway, it was not a successful run and must be redone.
8) After you have a successful run, write down the sequence of arrows on the reverse side of this page and do the next challenge.

The Challenges for your Robot

1) Create a program that gets the robot to move from the green arrow to the square marked with red. Ignore all other colors on the floor during this challenge question. Write your solution here:

2) Create a program that gets the robot to move from the green arrow to the square marked with red, but this time it cannot step on any square that is marked with silver. (Ignore the squares that are marked with pink, blue and orange, but stepping on a square marked with silver will cause the robot to stop moving immediately.) Write your solution to this challenge here:

3) Create a program that gets the robot to move from the green arrow to the square marked with red, but this time it must touch the pink, blue, and orange squares in that order on the way to the red square. (For this exercise, ignore the squares that are marked as silver squares). Write your solution to this challenge here:

4) Create a program that gets the robot to move from the green arrow to the square marked with red, touching the pink, blue, and orange squares in that order on the way. This time the robot will stop working if it steps on a square that is marked with silver. Write your solution to this challenge here:

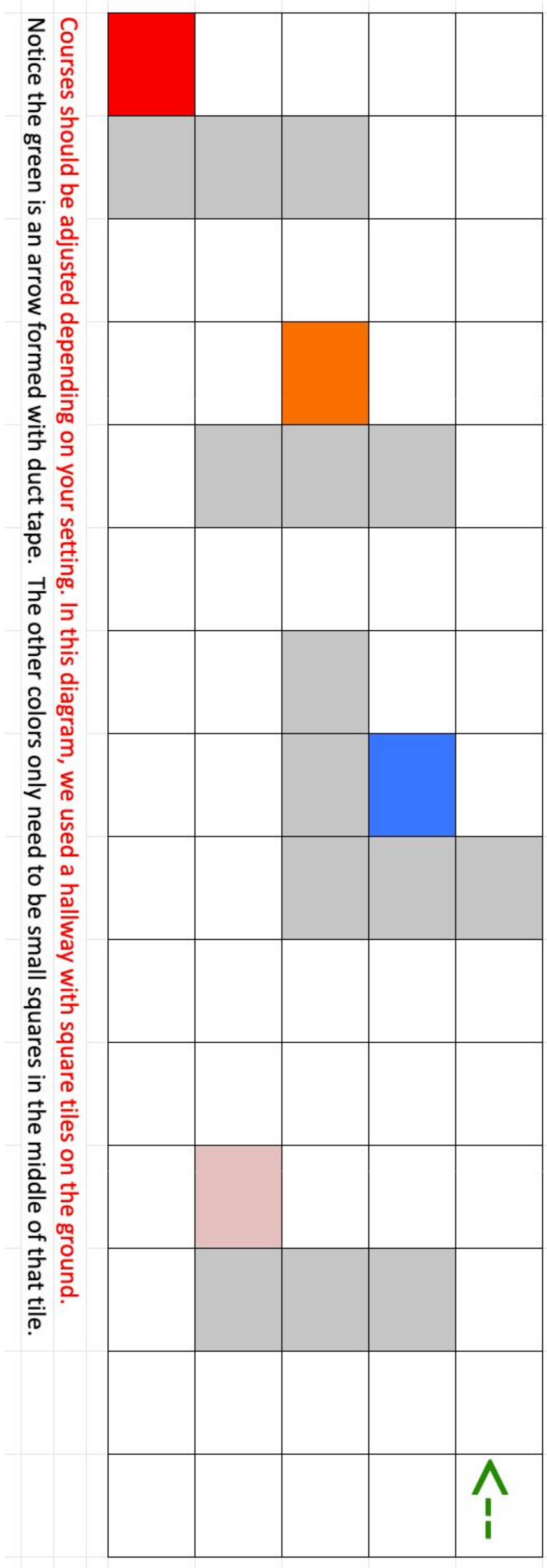
5) Create a program that gets the robot to move from the green arrow to the pink square then back to the green arrow, then moves from the green arrow to the blue square and then back to the green arrow, then moves from the green arrow to the orange square and back to the green arrow, and finally then moves to the red square. Through this whole process the robot must avoid silver squares. Write your solution here:

CT within Mystery STEM Careers

Below are job descriptions for selected STEM careers that require computational thinking along with some examples of how each career uses CT. Compare this list of CT practices to yours and add more ways CT is used in each of these careers.

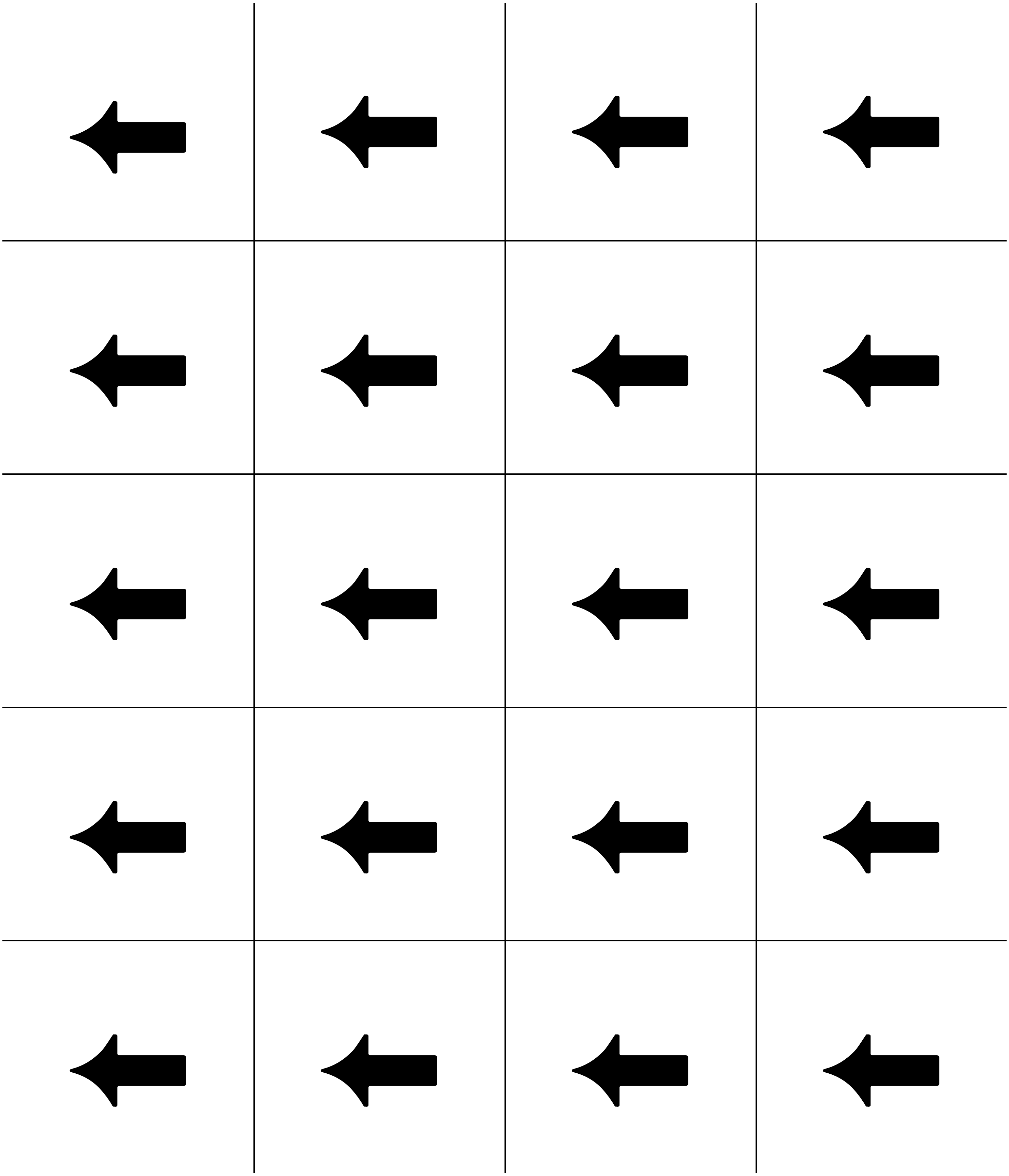
An Ecologist might…	• Study relationships within ecosystems • Collect and analyze data on environmental conditions • Use models to assess potential impacts of ecosystem changes • Address the problem of how to conserve biodiversity on our planet
An Astronomer might...	• Study how solar systems, galaxies, or the universe works • Use telescopes or other tools to collect images of stars and planets • Write computer programs to help analyze telescope images
A Bioengineer might...	• Analyze an organism as a system when a new factor is introduced • Use models and simulations to develop, test and modify products and software • Create an algorithm to determine specific conditions of an organism
A Mechanical Engineer might...	• Use CAD software to make initial design concepts based on measurements • Use models and simulations to develop, test and modify products • Determine how that new product will work with products already in place
A Medical Doctor might...	• Gather and analyze patients' vital signs (e.g., blood pressure, weight) • Determine which condition(s) may explain a person's symptoms • Design a treatment plan (e.g., prescriptions, surgery) • Prescribe a different antibiotic when antibiotic resistance is suspected
A Meteorologist might...	• Collect data about past events and current conditions • Use models and simulations to predict weather patterns • Put the data collected and likely future data into a user-friendly format • Determine how the various patterns are affected by the presence or absence of other patterns
An Actuarial Scientist might...	• Collect and use data about expected events • Use models and simulations to predict likelihood of those events • Determine how those events are likely to happen in the presence or absence of other events
A Mechanic might...	• Diagnose and troubleshoot issues with the function of a vehicle • Found cost effective solutions to issues • Recommend preventative measures to ensure the safety and function of a vehicle based on usage and reliability
A Plumber might...	• Follow plans to design and implement a water line system • Use said plans to diagnose and fix leaks and clogs to a system • Work to update and make revisions to existing water systems in a house or building

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct



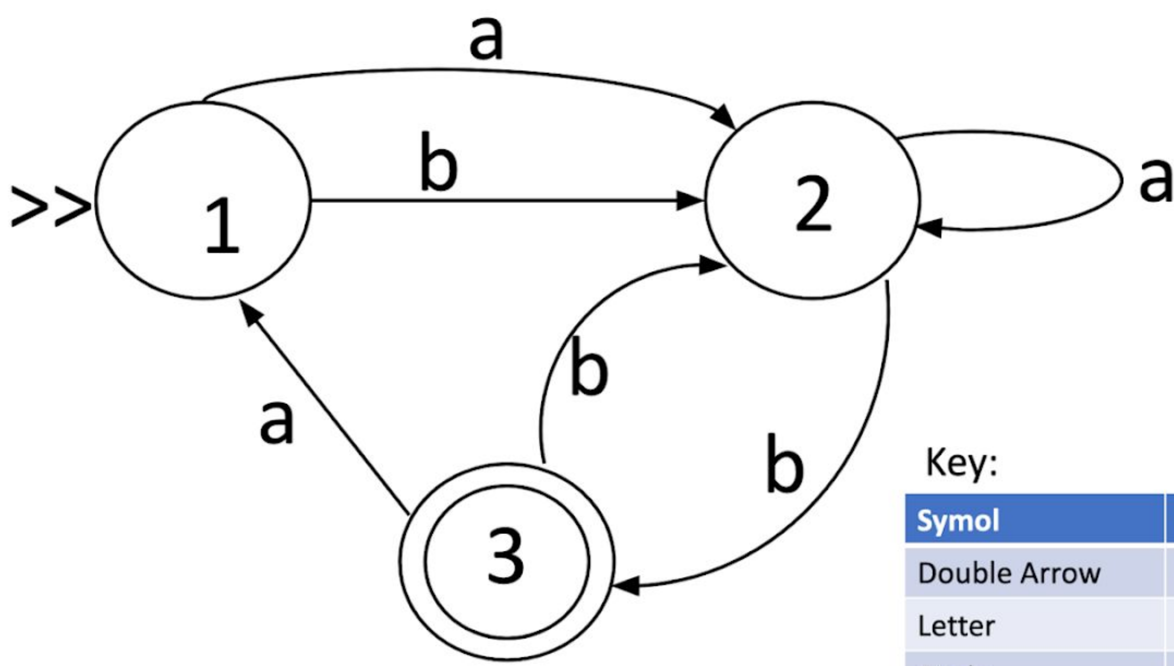
[Click here to access the Excel File]

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct



Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG^{CT}: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Station #1



Key:

Symol	Meaning
Double Arrow	Start state
Letter	Symbol in the alphabet
Circle	State of the finite state machine
Double Circle	End State of the machine
Numbers	Names for the States
Arrows	Direction of flow between states

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Station #1

Directions: Based on the words given, circle whether or not the word is accepted by the finite state machine. An accepted word is one that ends in a final state (double circle).

Word	Accepted? (Circle One)
aba	Yes No
bab	Yes No
abbab	Yes No
abababba	Yes No
bababa	Yes No

Give one word that would be accepted (one that isn't shown above):

Give one word that wouldn't be accepted (one that isn't shown above):

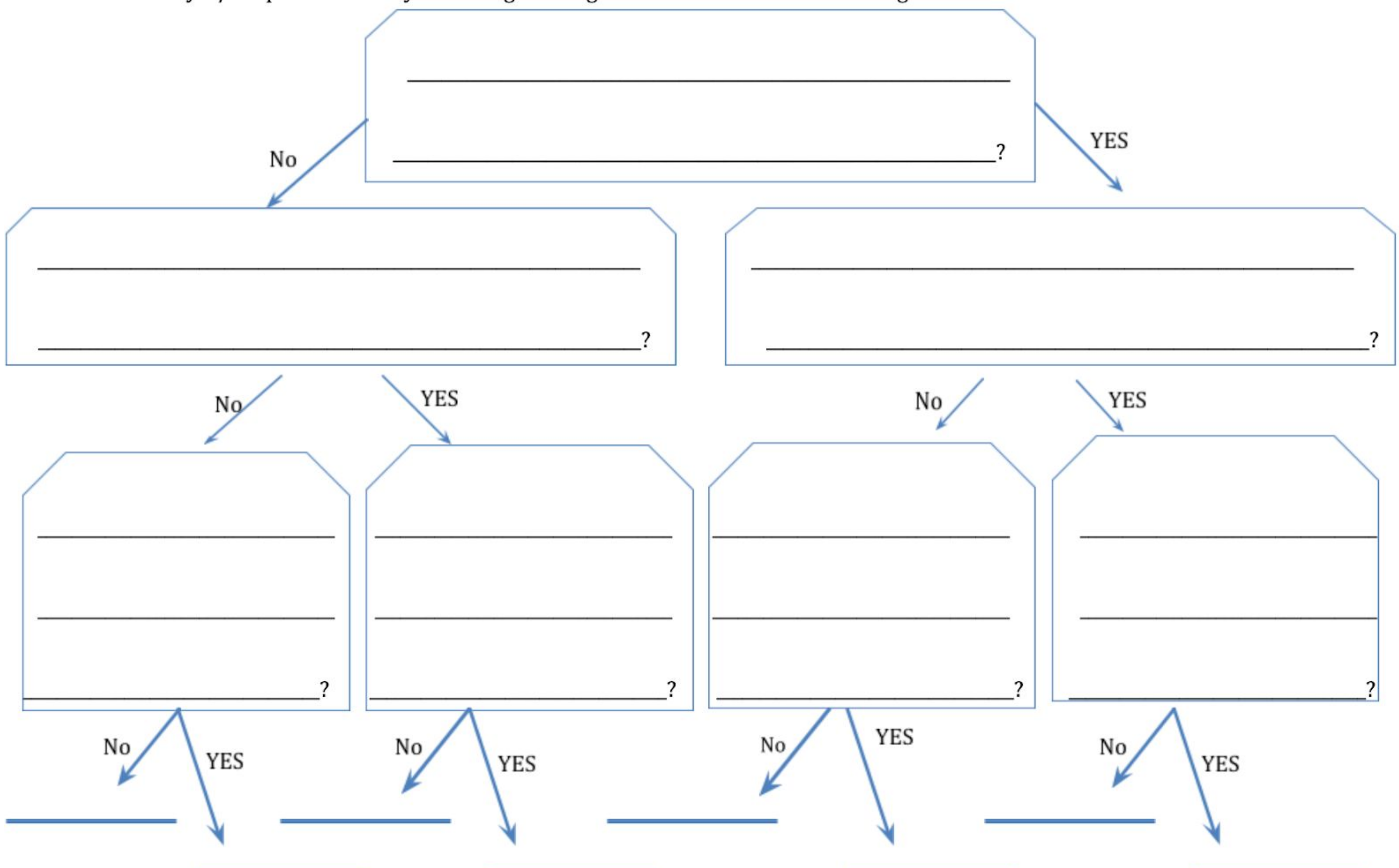
Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Station #2

 Tree	 Fish	 Bear	 Bird
 Flower	 Dolphin	 Tiger	 Rooster

Tree image by Tim Hill from Pixabay, Fish image by Robert Balog from Pixabay, Bear image by 358611 from Pixabay, Bird image by Timo Schlüter from Pixabay, Flower image by Spencer Wing from Pixabay, Dolphin image by Wolfgang Zimmel from Pixabay, Tiger image by Pexels from Pixabay, Rooster image by Pexels from Pixabay

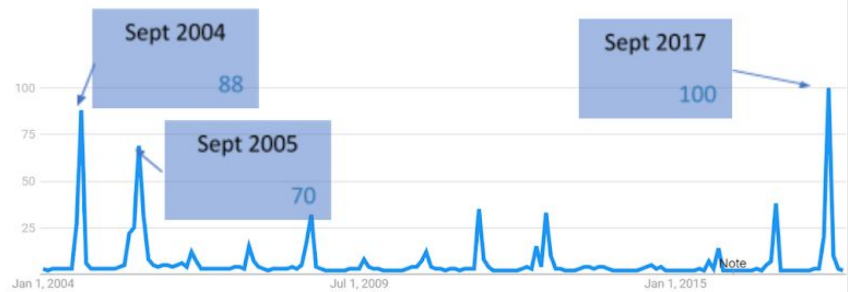
I am thinking of one of these organisms.
Show the tree of yes/no questions that you could give to figure out which one I am thinking of.



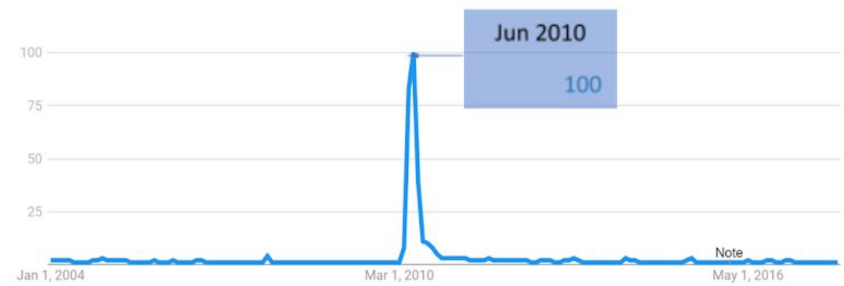
Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Station #3

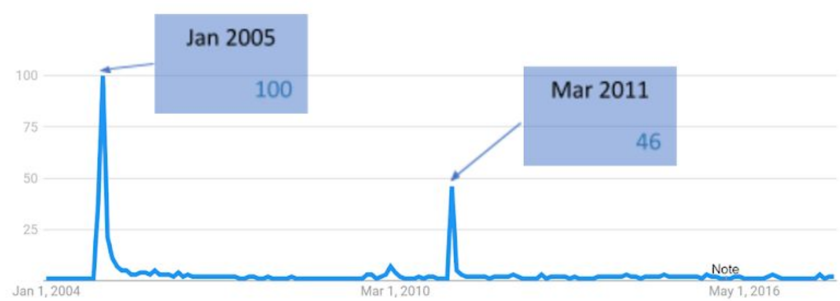
Directions: Match the graphs of the frequency of Google searches to the corresponding word on the table.



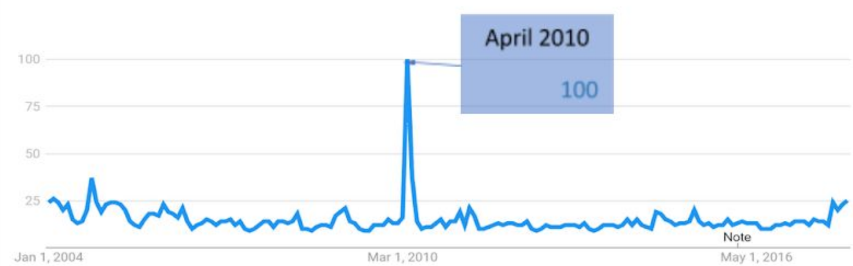
Word: ________________



Word: ________________



Word: ________________



Word: ________________

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Volcano	Oil Spill
Tsunami	Hurricane

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

Station #4

When data is compressed, it is put through a mask. A mask acts like a filter for the data, only letting some data through. Where the mask is a 1, the data goes through but where the mask is a 0, the data does not go through. This is shown in the graphic below. The 0's are marked in red to mean that they stop the data from falling through to the line of new data below, and 1's are marked in green to indicate the data element above falls through to the new data line.

1	1	1	1	0	0	0	data
1	0	0	1	1	1	0	mask
1	0	0	1	0	0	0	new data

Hexadecimal
0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111
8 = 1000
9 = 1001
A = 1010
B = 1011
C = 1100
D = 1101
E = 1110
F = 1111

You are going to get to experiment with masks. Go to:
https://example.pencilcode.net/edit/data-compression

On line #26 of the code on the screen, you see a long hexadecimal number. It starts with 0x and its initial value is 0xFFFFFFFF. This value indicates the mask.

Notice, when the mask is all F's in hexadecimal (meaning the binary is all 1's), the original data and the new data are exactly the same – the mask allows all data to fall through exactly as they are.

If any of those F's is changed to one of the other values shown in the hexadecimal conversion chart on the right, there will be some bits in the data that no longer go through the mask. This is because now some of the positions in the mask have the value 0 and prevent that bit from going into the new data.

Try each of the following values as a mask. Note the amount of compression and if the picture appears different to you or not. Fill out the form on the reverse side of this paper with your observations.

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct

What you should put in to replace the hexadecimal number on line 26 – then push the "run" triangle.	What is the new size of the picture file?	What do you see as a change in the picture?
0xFF000000		
0xFF0000FF		
0xFF444444		
0xFFF0F0F0		

Suggested Citation:
Coenraad, M., Plane, J., Ketelhut, D.J., Cabrera, L., Mills, K., Killen, H. (2020). STIG[CT]: The Science Teacher Computational Thinking Inquiry Group. Retrieved from https://education.umd.edu/stigct