Incrementing Numbers

Improve number sense by learning to increment numbers. Challenge students to find ways to increments by amounts more than one and less than one. Extend learning to consider negative numbers, Roman Numerals and more!

Created by: Susan Miller, University of Colorado, School of Education

This curricula has been designed as part of the Scalable Games Design project.

This material is based upon work supported by the National Science Foundation under Grant No. DRL-1312129 and CNS-1138526. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Incrementing Numbers (Continued)

Lesson Objective:

- **To strengthen number sense by understanding the process and rules when incrementing numbers**

Prerequisite Skills:

- **Students are presumed to know the following skills. Return to the Frogger Lesson Plans for detailed instructions on these skills.**
- **Create agents**
- **Basic agent behavior including:**
- **Key control**
- **Random movement**
- **Ending the game**

Computational Thinking Patterns:

- **Collision**

Length of Activity:

- **Two 30-45 minute lessons, although some students may advance more quickly**

Activity Description:

- **Part 1: Increment Numbers**
- **Part 2: Extend Incrementing Numbers**

Table of Contents

Vocabulary/Definitions

Collision .....the situation when two agents physically collide.

Depiction.....a second image of the original agent. For example, the Sokoban can have two depictions: what it usually looks like, and what it looks like after it has been squished

Increment.....to increase by one

Method.....a set of rules to follow in a specific situation

Incrementing Numbers

General Teaching Strategies[1]

Basic Philosophy

- The educational goal of these lessons is to learn and apply Computational Thinking Patterns in the context of a familiar game. Emphasis on these Computational Thinking Patterns is essential for student understanding.

- Every effort has been made to create instructions with an eye toward guided discovery. Direct instruction has been used for those aspects where students are learning the code for the first time; however, materials have been provided to ensure that students are understanding the programming concepts, as opposed to simply copying code. Note that special materials have been designed for students who are new to this program.

- Student materials are available for each portion of the game design. These materials are intended to be used in addition to teacher materials, which provide prompts and discussion points. Students may become frustrated with too little teacher support. Students may lose out on conceptual understanding with too much teacher support.

Guided Discovery Process

- **Model the process** rather than just giving students the answer. Building the game on your own, before trying it with your students will enable you to see possible struggling points.

- Have students work through problems on their own. Ask directing questions or give helpful suggestions, but **provide only minimal assistance** and only when needed to overcome obstacles.

- Don't fear **group work**! It is common for computer programmers to talk through problems with one another, and to use code snippets found from other programs, and other programmers. Talking through coding problems enables students to think more

[1] This information is supported by research found in the following documents:

Basawapatna, A. R., Koh, K. H., & Repenning, A. (2010, June). Using scalable game design to teach computer science from middle school to graduate school. In Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (pp. 224-228). ACM.

National Research Council. (2011). *Learning science through computer games and simulations.* (M. Hilton & M. Honey, Eds.). Washington, DC: The National Academies Press.

National Research Council. (2014). *STEM Integration in K-12 Education:: Status, Prospects, and an Agenda for Research*. (M. Honey, G. Pearson, & H. Schweingruber, Eds.). Washington, DC: The National Academies Press.

Repenning, A., & Ioannidou, A. (2008, March). Broadening participation through scalable game design. In *ACM SIGCSE Bulletin* (Vol. 40, No. 1, pp. 305-309). ACM.

critically about Computational Thinking Patterns, as well as the steps needed to solve a problem. Additionally, seeing how others solved an issue with code helps students realize that problems often have multiple solution strategies, and some that might be more effective than others

- Recognize that programming is largely a process of **trial and error**, particularly when first learning. It is helpful to encourage this mindset with your students.

Building Blocks

- Each project is designed to build on the prior one. Very little student support is provided where expertise has already been created. Conversely, material that is new has more support.

- Be sure to talk through the building blocks (especially for PacMan in the area of diffusion and hill climbing) as these Computational Thinking Patterns will appear often in future games and simulations.

- Remember that conceptual understanding takes time, and it may be necessary to explain these concepts multiple times, using different examples, so that all students can be successful.

Support Learning

- Research shows that game design is associated with engaged students, and engaged students show higher levels on conceptual understanding. Allowing students to personalize their games aids in this engagement and motivation.

- Coding may be difficult for some students, and all students are likely to be frustrated at times when the code does not produce the expected results. **Praise students** for sticking with the troubleshooting process and encourage them to share what they learned with others.

- Be sure to communicate that **the process is more important than the answer**, and that coding of a project often takes time. Do not place pressure on your students to 'hurry up' and resort to giving them the code. The process of figuring it out on his/her own will result in much stronger conceptual understanding.

Lesson Overview

While this lesson will enable your students to successfully use numbers in a variety of game situations, it has been designed and written to enhance student comprehension of how numbers are incremented. As a result, there are no student handouts associated with this lesson, although code for numbers is included in the Code Snippets lesson available online. This lesson is designed to be an interactive discussion between the teacher and the students.

> Note: While having the students all program their own number counting systems is recommended, this lesson could be done entirely on the teacher's computer using an overhead projector.

The goal of this lesson is to strengthen students' conceptual understanding of how our numbering system works. This can move from basic conceptual understanding (1 → 2, 2 → 3… to a more rich and complex understanding… What does a base 10 number system mean? How would this vary if it were written in Base 2? How would it change if we used Roman Numerals? Can we add parts of numbers in similar ways? Can we count backwards? The extensions go on and on.

Using technology to teach this topic will strengthen student conceptualization of our number system, ultimately strengthening number sense. As this is an entirely new way of thinking about how numbers increase, allow students to take time to talk through the problems, and to discuss their thought-process with others in the class for greater comprehension[2].

[2] Additional details of prior research in this area can be found in the following resources:

Judson, E. (2009). Improving technology literacy: does it open doors to traditional content? *Educational Technology Research and Development*, *58*(3), 271–284. doi:10.1007/s11423-009-9135-8

National Research Council. (2014). *STEM Integration in K-12 Education:: Status, Prospects, and an Agenda for Research*. (M. Honey, G. Pearson, & H. Schweingruber, Eds.). Washington, DC: The National Academies Press.

Teacher Instructions Part 1: Incrementing Numbers

Use paper to create two sets of digits from 0 through 9. Tell the students you are a robot and will be keeping score for a game. Ask the students to describe how to count numbers. Make sure they give you RULES and not DIRECTIONS. For example, they shouldn't say "replace that card with a 4", but rather "every time you see a four in the ones place value location, it increases by changing it to a 5".

Consider these prompts:

- Pretend I am a robot keeping score for a game. I have a board which holds numbers. How would you program me to increase the number by 1?
- What if I already had a number on the board? What it the current score was 5…what would I do? What if the current score was 0? What if the current score was 9?
- How do I continue to increase numbers? What if the current score was 29? 59? How can I program it for all circumstances?

Push student to *justify* their answers. How do they know their rule works? Does it work for all situations? Does it work for all place values?

There is a tremendous amount of mathematical learning that comes from the conceptual understanding of how numbers grow. Give your students time to talk through <u>and</u> think through the process.

Eventually, the rules should generally look like this:

- 0 → 1
- 1→2
- 2→3
- 3→4
- 4→5
- 5→6
- 6→7
- 7→8
- 8→9
- 9→0 and the number of the left increases by 1 (e.g. 29 → 30…the nine changes to a zero, and the 2 increments by one to a 3)
- 9→0 and a new number 1 is added to the left (for the case from 9 → 10, where there is no current digit to the left)

Then push their understanding even further: Pretend I am a robot keeping score for a game. I have a board which holds numbers. How would you program me to increase the number by 5? By 4? By 10? What would the rules look like? Next, ask the students how to program this in

Incrementing Numbers (Continued)

AgentSheets… how many agents will they need? Who will act as the robot? How will the robot know it's time to increment the number? What changes if we want to increment by 3? By10?

Agents:

- Create an Agent called DIGITS. Then, create depictions within that agent that represent the numbers 0-1-2-3-4-5-6-7-8-9
- Create an Agent called Game Master (or Robot or Controller…)

Tips and Tricks

We actually do not need to modify the original Letter or Number depiction since we will not use it in our game, however it can be helpful to modify it.

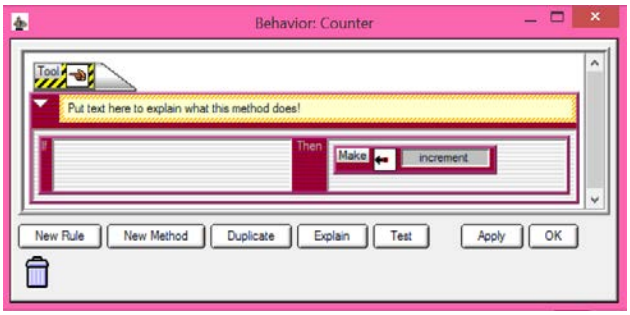
Change the MASK color to WHITE to make it transparent!

Save an interesting background image to the computer – use that as the background for this counter. You can load a background by clicking on the worksheet, and then on FILE>>Load Background.

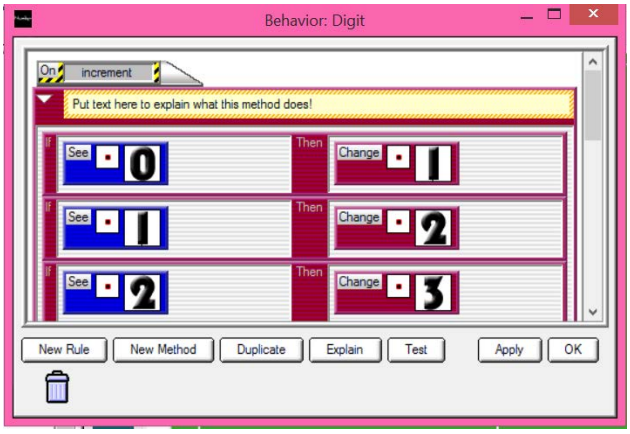
Incrementing Numbers (Continued)

Agent Behaviors:

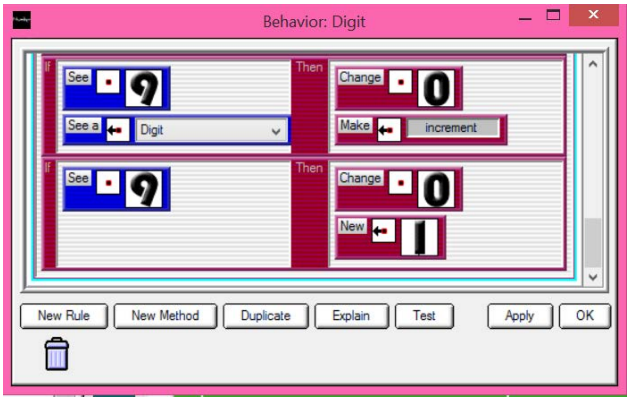
First – tell the controller to make the number increment. In this case, we used a special method using the tools…the method is called when the user clicks on the controller using the finger tool.



Then – tell the number HOW to increment. The first three are shown…the rest must be added.



Note – there are special rules for what happens when there is a nine in the one's digit!



We used the "See A" Action for one of the rules. The difference between the "See" and "See A" Actions is that "See A" looks for <u>any</u> Agent regardless of the Depiction, while the "See" Action looks for a <u>specific</u> Depiction of an Agent.

We could program the same behavior using the "See" action as we did using the "See A" action, however it would require a separate rule for each Numbers depiction!

Teacher Instructions Part 2: Challenges

What needs to change in the rules to count by 3's…5's…10's?

How would we program this?

What needs to change in the rules to count by decimals?

How would we program this?

What needs to change in the rules to count backwards?

How would we program this?

What needs to change in the rules to include negative numbers when we count backwards?

How would we program this?

What needs to change in the rules to count in Roman Numerals?

How would we program this?

How can we use this in a game to…

- **Keep score?**
- **Count steps?**
- **Other ideas??**

Incrementing Numbers

ISTE Standards[3] specific to the implementation of Incrementing Numbers (Denoted with (✸))

Creativity and Innovation

Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:

Apply existing knowledge to generate new ideas, products, or processes:

- Design and develop games
- ✸ Design and develop computational science models

Create original works as a means of personal or group expression.

- Design original games
- Model your local environment, e.g., ecology, economy

Use models and simulations to explore complete systems and issues.

- Model scientific phenomena, e.g., predator / prey models
- ✸ Create visualizations

Identify trends and forecast possibilities.

- Build predictive computational science models, e.g., how the pine beetle destroys the Colorado pine forest
- Build live feeds to scientific web pages (e.g, weather information), process and visualize changing information

Communication and Collaboration

Students use digital media and environments to communicate and work collaboratively, including at a distance, to support individual learning and contribute to the learning of others. Students:

Interact, collaborate, and publish with peers, experts, or others employing a variety of digital environments and media:

- ✸ Students work in teams to build and publish their simulations as web pages containing java applets.

Communicate information and ideas effectively to multiple audiences using a variety of media and formats.

- Effectively combine interactive simulations, text, images in web pages

Develop cultural understanding and global awareness by engaging with learners of other cultures.

- ✸ Students and teachers from the four culturally diverse regions interact with each other

Contribute to project teams to produce original works or solve problems.

- ✸ Define project roles and work collaboratively to produce games and simulations

[3] ISTE Standards for Students (ISTE Standards•S) are the "standards for evaluating the skills and knowledge students need to learn effectively and live productively in an increasingly global and digital world." http://www.iste.org/standards/standards-for-students

Incrementing Numbers (Continued)

Research and Information Fluency

Students apply digital tools to gather, evaluate, and use information. Students:

Plan strategies to guide inquiry.

Explore web sites and identify interesting connections

Locate, organize, analyze, evaluate, synthesize, and ethically use information from a variety of sources and media.

Find relevant related web-based information, compute derivate information

Evaluate and select information sources and digital tools based on the appropriateness to specific tasks.

Understand validity of information, e.g. Scientific journal information vs. Personal blogs

Process data and report results.

Write programs to access numerical information, define functions to process data and create output based on voice or plotting to represent data.

Critical Thinking, Problem Solving, and Decision Making

Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources. Students:

Identify and define authentic problems and significant questions for investigation.

Define research questions and explore approach of exploration

Plan and manage activities to develop a solution or complete a project.

- Outline sequence of exploratory steps
- Experience complete bottom-up and top-down design processes
- Employ algorithmic thinking for creating programs to solve problems

Collect and analyze data to identify solutions and/or make informed decisions.

Collect data as time series, e.g., collect group size of predator and prey, export time series to excel, explore various types of graph representations, e.g., x(t), y(t) or scatter y=f(x)

Use multiple processes and diverse perspectives to explore alternative solutions.

- Experience and understand design trade-offs, e.g. Bottom-up vs. Top-down

Digital Citizenship

Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior. Students:

Advocate and practice safe, legal, and responsible use of information and technology.

- Learn how to use tools to locate resources, e.g., images with google image search, but understand copyright issues

Exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity.

- Stay in the flow, where design challenges match design skills
- Experience success through scaffolded game design activities

Incrementing Numbers (Continued)

- Mentor other students

Demonstrate personal responsibility for lifelong learning.

- Explore options of going beyond expected learning goals

Exhibit leadership for digital citizenship.

- In a collaborative setting become a responsible producer of content for diverse audiences

Technology Operations and Concepts

Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:

Understand and use technology systems.

- Know how to organize files and folders, launch and use applications on various platforms

Select and use applications effectively and productively.

- Know how to orchestrate a set of applications to achieve goals, e.g., make game and simulations using Photoshop (art), AgentSheets (programming), and Excel (data analysis).

Troubleshoot systems and applications.

- Debug games and simulations that are not working

Transfer current knowledge to learning of new technologies.

- Reflect on fundamental skills at conceptual level. Explore different tools to achieve similar objectives.