2nd STEAM – Unit 2 (COMPUTER PROGRAMMING)

Content Area: **Computer Programming (Grade 2)**
Course(s):
Time Period: **Ongoing**
Length: **Ongoing**
Status: **Published**

Big Idea

Learning computer science encourages critical thinking, logic, persistence, and creativity. It can help students excel at problem-solving in all subject areas, no matter what their age. At the root of all computer science is something called an algorithm. The word "algorithm" may sound like something complicated, but really it's just a list of instructions that someone can follow to achieve a result. To provide a solid base for computer science education, students should build a secure relationship with algorithms. (CODE.ORG, 2017)

Enduring Understanding

SWBAT write algorithms to control a digital character's motion, sound, color, and response to stimuli. SWBAT identify patterns in code.

Skills

- Using digital block coding applications, students write algorithms to control a digital character and it's surroundings. Students write algorithms to control the digital character's motion, sound, color, and response to stimuli (cues).
- Solve problems using sequential algorithms, debugging algorithms, creating loops in algorithms, and creating conditional algorithms.
- Practice problem solving and perseverance techniques.

Standards

8.2.2.E.1 List and demonstrate the steps to an everyday task.
8.2.2.E.2 Demonstrate an understanding of how a computer takes input through a series of written commands and then interprets and displays information as output.
8.2.2.E.3 Create algorithms (a sets of instructions) using a pre-defined set of commands (e.g., to move a student or a character through a maze).
8.2.2.E.4 Debug an algorithm (i.e., correct an error).
8.2.2.E.5 Use appropriate terms in conversation (e.g., basic vocabulary words: input, output, the operating system, debug, and algorithm).

Assessments

- Tracking of student achievement in Code.org's (or other online coding application's) lessons.
- Teacher observation

Resources/Instructional Materials

CODE.ORG COURSE C. https://studio.code.org/s/coursec?section_id=1057489 .Create programs with sequencing,

loops, and events. Translate your initials into binary, investigate different problem-solving techniques, and learn how to respond to cyberbullying. At the end of the course, create your very own game or story you can share!

Lesson 15: Beyond Programming: Binary Bracelets

Binary is extremely important in the world of computers. The majority of computers today store all sorts of information in binary form. This lesson helps demonstrate how it is possible to take something from real life and translate it into a series of ons and offs. In this lesson students will learn how information is represented in a way such that a computer can interpret and store it. When learning binary, students will have the opportunity to write codes and share them with peers as secret messages. This can then be related back to how computers read a program, translate it to binary, use the information in some way, then reply back in a way humans can understand. For example, when we type a sentence into a document then press save, a computer translates the sentence into binary, stores the information, then posts a message indicating the document has been saved.

Lesson 2: Programming in Maze

Using characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of a maze to the pig at the other side. To do this they will stack code blocks together in a linear sequence, making them move straight, turn left, or turn right. In this lesson, students will develop programming and debugging skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

Lesson 3: Debugging in Maze

Debugging is an essential element of learning to program. In this lesson, students will encounter puzzles that have been solved incorrectly. They will need to step through the existing code to identify errors, including incorrect loops, missing blocks, extra blocks, and blocks that are out of order. Students in your class might become frustrated with this lesson because of the essence of debugging. Debugging is a concept that is very important to computer programming. Computer scientists have to get really good at facing the bugs in their own programs. Debugging forces the students to recognize problems and overcome them while building critical thinking and problem solving skills.

Lesson 5: Programming in Collector

In this series of puzzles, students will continue to develop their understanding of algorithms and debugging. With a new character, Laurel the Adventurer, students will create sequential algorithms to get Laurel to pick up treasure as she walks along a path. In this lesson, students will be practicing their programming skills using a new character, Laurel the Adventurer. When someone starts programming they piece together instructions in a specific order using something that a machine can read. Through the use of programming, students will develop an understanding of how a computer navigates instructions and order. Using a new character with a different puzzle objective will help students widen their scope of experience with sequencing and algorithms in programming.

Lesson 6: Programming in Artist

In this lesson, students will take control of the Artist to complete drawings on the screen. This Artist stage will allow students to create images of increasing complexity using new blocks like move forward by 100 pixels and turn right by 90 degrees. Building off of the students' previous experience with sequencing, this lesson will work to inspire more creativity with coding. The purpose of this lesson is to solidify knowledge on sequencing by introducing new blocks and goals. In this case, students learn more about pixels and angles using the new blocks, while still practicing their sequencing skills. Also, students will be able to visualize new goals such as coding the Artist to draw a square.

Lesson 8: Loops with Rey and BB-8

Building on the concept of repeating instructions from "Getting Loopy," this stage will have students using loops to help BB-8 traverse a maze more efficiently than before. In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using loops is an important skill in programming

because manually repeating commands is tedious and inefficient. With the Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be looped. It should be noted that students will face puzzles with many different solutions. This will open up discussions on the various ways to solve puzzles with advantages and disadvantages to each approach.

Lesson 9: Loops in Artist

Watch student faces light up as they make their own gorgeous designs using a small number of blocks and digital stickers! This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity is fantastic for producing artifacts for portfolios or parent/teacher conferences. This series highlights the power of loops with creative and personal designs. Offered as a project-backed sequence, this progression will allow students to build on top of their own work and create amazing artifacts.

Lesson 10: Loops in Harvester

In the preceding stage, students used loops to create fantastic drawings. Now they're going to loop new actions in order to help the harvester collect multiple veggies growing in large bunches. It may seem unnecessarily repetitive to have two plugged stages introducing loops, but the practice of using loops for different reasons develops a student's understanding of what loops can do. In "Loops in Maze" students only used loops to repeat movements. In this lesson, students will use loops to repeat other actions like harvesting pumpkins. New patterns will emerge and students will use creativity and logical thinking to determine what code needs to be repeated and how many times.

Lesson 12: Build a Flappy Game

In this special stage, students get to build their own Flappy Bird game by using event handlers to detect mouse clicks and object collisions. At the end of the level, students will be able to customize their game by changing the visuals or rules. Events are very common in computer programs. In this lesson, students will further develop their understanding of events by making a Flappy Bird game. Students will learn to make their character move across the screen, make noises, and react to obstacles based on user-initiated events.

Lesson 13: Events in Play Lab

n this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all the coding skills they've learned to create an animated game. It's time to get creative and make a game in Play Lab! Here, students will further develop their understanding of events using Play Lab. Students will use events to make characters move around the screen, make noises, and change backgrounds based on user input. At the end of the puzzle sequence, students will be presented with the opportunity to share their projects.

WEB SITES

- Code.org lesson plans
- Scratch lesson plans
- Tynker lesson plans

Modifications

Individual accommodations

• Additional support

• Adapting lessons to meet various learning styles

Integration of 21st Century Skills

Focus on the development of 21st Century Content Skills:

- Global awareness
- Civic literacy
- Health and wellness awareness
- Environmental literacy

Focus on the Development of Learning and Thinking Skills:

- Critical Thinking and Problem Solving Skills
- Communication Skills
- Creativity and Innovation Skills
- Collaboration Skills
- Information and Media Literacy Skills
- Contextual Learning Skills

Focus on the Development of Life Skills:

- Leadership
- Ethics
- Accountability
- Adaptability
- Personal Productivity
- Personal Responsibility
- People Skills
- Self Direction
- Social Responsibility

Interdisciplinary Connections

- Academic and Technical Rigor - Projects are designed to address key learning standards identified by the school or district.
- Authenticity - Projects use a real world context (e.g., community problems) and address issues that matter to the students.
- Applied Learning - Projects engage students in solving problems calling for competencies expected in high-performance work organizations (e.g.,teamwork, problem-solving, communication, etc.).
- Active Exploration - Projects extend beyond the classroom by connecting to community explorations.
- Adult Connections - Projects connect students with the wider community.
- Assessment Practices - Projects involve students in regular, performance-based exhibitions and assessments of their work; evaluation criteria reflect personal, school, and real-world standards of performance.