



INTRO TO COMPUTER SCIENCE: VISUAL PROGRAMMING

Grade Level: 2nd Duration: 2-3 Class Periods

PROJECT DESCRIPTION



Welcome to Scratch!

In this computer science project, we will explore visual programming through Scratch. Scratch is a user-friendly platform that let students create interactive stories, games, and animations. It's both fun and a great introduction to coding. You will follow the "I do, We do, You do, teaching approach: first, the teacher demonstrate a simple project; next, the teacher and students work together, troubleshooting with support from each other, finally, the students will have the chance to try it out on their own. Throughout the project, we'll focus on creativity, problem-solving, teamwork, and collaboration. By the end, students will be amazed at what they can create!

There are two ways we can use Scratch: by downloading the installation file from <https://scratch.mit.edu/download> and or by working <https://scratch.mit.edu/>. Both ways have their advantages and disadvantages.

If your school has a computer lab, it's best to install the desktop version on the computers in the lab, the program will **run faster** and more smoothly, **does not require an internet connection**, and can be customized to fit the needs of the classroom. However, the desktop version does not have as many features as the online version, such as the ability to collaborate with other students or create multiplayer games.

Whereas the online platform provides the ability for **students to collaborate** and create multiplayer games. Requires no installation or setup, always up-to-date with the latest features, and students can **access their projects from any device** with an internet connection. However, it may run slower, especially if there are many students using it at the same time. Requires an internet connection, and students may not be able to save their projects if the internet connection is interrupted. Ultimately, the best version of **Scratch to use will depend on your specific needs and preferences**.

STANDARDS

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming:

- 1A-AP-11 - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
- 1A-AP-09 - Model the way programs store and manipulate data by using numbers or other symbols to represent information.
- 1A-AP-10 - Develop programs with sequences and simple loops, to express ideas or address a problem.
- 1A-AP-11 - Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
- 1A-AP-14 - Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.



START CODING IN SCRATCH

GOALS AND OBJECTIVES

Goal:

- By the end of this lesson, students will be able to demonstrate an understanding of basic Scratch programming concepts by effectively using Control, Motion, Looks, and Sound blocks to create a simple interactive animation.

Objectives:

- Students will identify and describe the purpose of Control, Motion, Looks, and Sound blocks.
- Students will apply their understanding of these blocks to create a Scratch project.
- Students will collaborate and share their projects, reflecting on the creative choices they made.



VOCABULARY

- Block Codes - are color-coded and explain what each of the blocks in Scratch does. i.e. Blue is motion.
- Programming Blocks - blocks that program the sprite to perform a specific task.
- Script: A series of code blocks that tell a sprite what to do.
- Sprite: A graphic that can be moved around the stage and programmed to do things
- Stage: here backdrops, sprites, costumes, and sounds are displayed.
- Command block: A code block that tells a sprite to do something, such as move, jump, or say something.
- Control Blocks: Special blocks that help tell the computer what to do first, like "start" or "if."
- Motion Blocks: Blocks that make characters move around, like "go" or "move."
- Looks Blocks: Blocks that change how things look, like giving characters new clothes.
- Sound Blocks: Blocks that make the computer play sounds.



IMPLEMENTING THE 5ES

Scratch is a visual programming language; therefore it is best taught in front of a computer running Scratch. In this lesson, students will learn the basic components of the Scratch interface, including sprites, the stage, and code blocks. Students will learn how to animate a sprite and see it come to life! As students explore and learn how to code, they are building skills that develop a strong foundation in problem-solving, creativity, and collaboration. There's no better way to dive into the exciting world of STEM! Topics Covered in this lesson include Building Blocks to Programming and the use of Control, Motion, Looks, and Sound blocks. Before getting started in any scratch activity, [CLICK HERE](#) to take a look at the **Getting Started with Scratch** video first.

START CODING IN SCRATCH



Engage (Duration: 15 minutes)



- Gaming is a passion for students. A fascinating discussion to ignite their enthusiasm revolves around the games they are currently downloading and playing. Include a discussion about students creating their own games and explore game creation programs they may already be familiar with, such as Minecraft, Tynker, Code.org, Scratch Jr, Scratch, etc. Since most students have experience with gaming, they can easily relate to playing or developing a game. Let's move on to some engaging questions that will help them to realize that they can also create their own games and animations.
- Here are some specific questions you can ask students to get them talking about gaming:
 - What are your favorite games?
 - What do you like about those games?
 - What do you dislike about those games?
 - What would you change about those games?
 - Have you ever tried to make your own game? If so, what was it like?
 - If you could make any game, what would it be?



Explore (Duration: 30 minutes)

- Have students go to Scratch.edu and create an account. If the student has an account, have them sign in.



[CLICK HERE TO FOLLOW THE JOIN SCRATCH VIDEO](#)

- Next, present a live demonstration of the Scratch interface, highlighting the Stage, Sprites, Script, and Programming palette. Explain the 4 main elements and how they interact. (**[CLICK HERE TO Review the "Scratch Interface Slides"](#)**).
- Follow the step-by-step instructions on the **Intro to Animation document** and guide students through creating a script using Control, Motion, Looks, and Sound blocks. demonstrate how to create an animation project, integrating the Control, Motion, Looks, and Sound blocks.
- Have students explore the blocks by allowing them to drag blocks onto the script area to move the cat across the screen, change his looks, etc. (allow students to explore, do not set parameters.)
- Observe students' creativity and answer questions related to the Control, Motion, Look, and Sound blocks only. This is unless your class is set up for differentiated instruction or you are comfortable with students progressing through the lesson.



Explain (Duration: 20 minutes)

- Use the intro to animation sheet as a guide to introduce the concept of programmable blocks: Control, Motion, Looks, and Sound.
- Demonstrate how to drag and connect blocks to make characters move, change appearance, and produce sounds by presenting a simple scenario (e.g., a dancing cat).
- Explain the concepts of programmable blocks and their categories: Control, Motion, Looks, and sound.
- Walk through the process step by step, explaining how each type of block contributes to the animation.
- Demonstrate how to drag and connect blocks to make characters move, change appearance, and produce sounds.



Elaborate (Duration: 40 minutes)

- Divide students into pairs or small groups.
- Provide each group with a scenario from the intro to animation sheet, and challenge them to collaboratively create an animation using the blocks they've learned.
- Encourage experimentation and creativity, allowing groups to explore different combinations of blocks.



Evaluate (Duration: 15 minutes)

- Ask each group to present their animations to the class.
- Have them explain their creative choices and how they used Control, Motion, Looks, and Sound blocks in their project.
- Assess each group's understanding of the block concepts through their presentations and the effectiveness of their animations.



Extend:

Encourage students to enhance their projects by adding more complexity or interactions.



CHECK FOR UNDERSTANDING

- ✦ **Observations during the "Elaborate" phase to assess how well students collaborate and apply their understanding of the different blocks.**
- ✦ **Quality and creativity of the animations presented by each group.**
- ✦ **Use the rubric to grade students work.**

INTRO TO ANIMATION SCRIPT

Guide students through creating a script using Control, Motion, Looks, and Sound blocks.

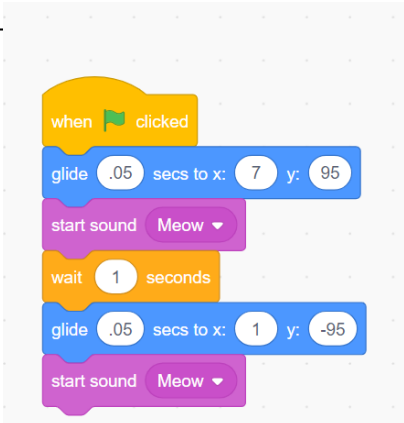
Explore with this simple animation script that makes a sprite jump up and down on the screen, without repeating.

- When green flag clicked
- Jump Up
- Wait 1 seconds
- Go back down
- Make a sound

No variables were added, explore with sprite selections, movement, and sound as you like.

However, in the sample script below, the script will first make the sprite move to the coordinates (7, 95), make a sound, wait for 1 second, return to bottom of stage, then make a sound.

T MART BOARD:



```

when green flag clicked
  glide .05 secs to x: 7 y: 95
  start sound Meow
  wait 1 seconds
  glide .05 secs to x: 1 y: -95
  start sound Meow
  
```

Have the students to experiment with different blocks to create your own animations. Here are some ideas and scenarios:

- Make a sprite walk across the screen.
- Make a sprite dance.
- Make a sprite tell a story.
- Make a sprite play a game.

The possibilities are endless!