Animate My Story in Scratch

Course Overview

Driving Question
How can you use coding and creativity to animate a story in Scratch?
In this course, coders are challenged to use Scratch to design and build an animated story. Students will learn the essential components of good storytelling, the basics of the Scratch platform, and how to translate creative ideas from brainstorming to coding. Students will leave with an understanding of fundamental computer science concepts like sequencing, loops, and events.

Course Layout
Modules
Module 1: Introduction to Boolean Girl Module 2: Storytelling 101 Module 3: Introduction to the Scratch Platform Module 4: Plan a Story Module 5: Coding in Scratch Module 6: Loops Module 7: Endings Module 8: Gallery Walk and Debugging Module 9: Presentations
Other Collateral
Activities: • Activity: Gallery Walk • Activity: Three Things • Activity: Block Game • Activity: PB & J Worksheets: • Worksheet: My Story Plan (Blank) • Worksheet: My Story Plan (Complete)
Schedules
5-Day Camps

Learning Targets
As a Boolean Girl, I can:

- ➔ Write a story and outline its main components
- ➔ Define terms such as computer, computer science, and coding
- ➔ Identify Scratch as a programming language.
- ➔ Plan code on paper
- ➔ Establish a stage and sprite(s) for my Scratch program
- ➔ Explain core CS concepts like loops and events
- ➔ Animate a complete story using Scratch
- ➔ Debug my code so it runs smoothly

Character and Community Building

Attributes embedded in this course:

- Communicator
- Collaborator
- Creative and Critical Thinker
- Goal-Directed and Resilient Individual

Vocabulary

- ➔ **Computer -** an electronic device that can be programmed with a series of instructions.
- ➔ **Coding/Programming -** providing instructions for a computer to follow. We use programming languages to code.
- ➔ **Programmer/Coder -** the person that writes the program or code.
- ➔ **Computer Science -** the study of computers and computing.
- ➔ **Scratch -** a block-based programming language that we will use in this course.
- ➔ **Character** – a person, animal, thing, or other being in a story. A main character is the character that is most central to the story. For example, in Cinderella, Cinderella is the main character.
- ➔ **Setting** – where the story takes place. For example, Frozen takes place in the kingdom of Arendelle, a cold and icy place.
- ➔ **Plot** – what happens in a story. For example, in Toy Story, a group of toys come alive and have adventures.
- ➔ **Beginning:** the first part of the story. This is a great place to get the reader/viewer's attention. It might include describing the characters or setting.
- ➔ **Middle:** most of the plot is in the middle. It might include interaction between characters or introducing some kind of problem.
- ➔ **End:** how we finish the story. It could include characters learning an important lesson or solving a problem. A great story has to have a great ending.
- ➔ **Block:** Each block contains a specific instruction for the computer. In Scratch, we code by putting blocks together to make a list of instructions.
- ➔ **Block Palette:** This is where we get the blocks we'll use.
- ➔ **Sprite:** A sprite is an image or object in our program. Sprites will follow our instructions.
- ➔ **Sprite Pane:** Shows us which sprites we currently have in the program.
- ➔ **Stage:** This is where we watch our program when it runs.
- ➔ **Backdrop:** A scene that we can add to the stage as a background for a program.
- ➔ **Scripts Area:** This is where we put our code blocks. Show students how to drag blocks into the scripts area.
- ➔ **Script:** a set of instructions that a programmer gives to a computer.

- ➔ **Event:** A trigger that makes something happen in a program. In our programs, we use events to tell the program when to start. We always start our scripts with an event.
- ➔ **Loop:** a computer science concept that makes a program do something over and over again. In Scratch, the loop blocks are `repeat`, `repeat until`, and `forever`.
- ➔ **Bug:** a mistake or error in code.
- ➔ **Debugging:** finding and fixing mistakes in code.
- ➔ **Feedback:** advice or reactions to someone's work.
- ➔ **Glows:** Positive feedback. Something you love about a person's work.
- ➔ **Grows:** Constructive feedback. Something that could improve in someone's work.

Materials and Preparation	
Traditional Classroom	**Virtual Classroom**
Course Overview Suggested Schedules Module Teacher Guides Slides Sample Projects Computer/Projector Paper and Writing Utensils Post-it Notes Beach Ball Boolean Boxes	Course Overview Suggested Schedules Module Teacher Guides Slides Sample Projects Computer Paper and Writing Utensils

Standards	
Virginia Computer Science Standards of Learning	
3.1	The student will construct sets of step-by-step instructions (algorithms), both independently and collaboratively a) using sequencing; b) using loops (a wide variety of patterns such as repeating patterns or growing patterns); and [Related SOL: Math 3.16] c) using events.
3.2	The student will construct programs to accomplish tasks as a means of creative expression using a block or text based programming language, both independently and collaboratively a) using sequencing; b) using loops (a wide variety of patterns such as repeating patterns or growing patterns); and c) identifying events.
3.3	The student will analyze, correct, and improve (debug) an algorithm that includes sequencing, events, and loops. [Related SOL areas – Math: Problem Solving, English: Editing]

3.4 The student will create a plan as part of the iterative design process, independently and/or collaboratively using strategies such as pair programming (e.g., storyboard, flowchart, pseudo-code, story map. [Related SOL: English 3.8c]

3.6 The student will break down (decompose) a larger problem into smaller sub-problems, independently or collaboratively. [Related SOL: Math 3.3b]

3.7 The student will give credit to sources when borrowing or changing ideas (e.g., using information and pictures created by others, using music created by others, remixing programming projects). [Related SOL: English 3.10e]

3.9 The student will identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., rebooting the device, checking for power, checking network availability, closing and reopening an app).

CSTA K-12 Standards

1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies**.**

1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.

1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.

1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.