Unit 3: Interactive Animations and Games

Content Area: **Business**
Course(s):
Time Period: **Semester 1 & 2**
Length: **3 weeks**
Status: **Published**

Standards

8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
	Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.
8.1.12.AP.7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.

Essential Questions

Chapter 1 - Images and Animations

- What is a computer program?
- What are the core features of most programming languages?
- How does programming enable creativity and individual expression?
- What practices and strategies will help me as I write programs?

Chapter 2 - Building Games

- How do software developers manage complexity and scale?
- How can programs be organized so that common problems only need to be solved once?
- How can I build on previous solutions to create even more complex behavior?

Enduring Understanding

- Create an interactive animation or game that includes basic programming concepts such as control structures, variables, user input, and randomness.
- Work with others to break down programming projects using sprites and functions.

- Give and respond constructively to peer feedback, and work with teammates to complete a project.
- View yourself as a computer programmer, and see programming as a fun and creative form of expression.

Knowledge and Skills

- Students learn fundamental programming constructs and practices in the JavaScript programming language.
- Students develop animations and games in Code.org's Game Lab environment.
- Students design their own animations and games by the end of the unit.

Transfer Goals

Students build on their coding experience as they create programmatic images, animations, interactive art, and games. Starting off with simple, primitive shapes and building up to more sophisticated sprite-based games, students become familiar with the programming concepts and the design process computer scientists use daily. They then learn how these simpler constructs can be combined to create more complex programs.

Resources

- Web Lab - Programming Environment
- Code.org Videos
- Code.org Demonstration Apps
- Code.org Activity Guide