SPRING GROVE AREA SCHOOL DISTRICT

PLANNED COURSE OVERVIEW

Course Title: Game Programming and Design	**Length of Course:** 15 cycles
Grade Level(s): 10-12	**Periods Per Cycle:** 6
Units of Credit: .5	**Length of Period:** 43 minutes
Classification: Elective	**Total Instructional Time:** 64.5 hours

Course Description

This course teaches students how to develop games using both Adobe Flash, Alice 2.0, and additional industry driven resources. Students will learn the basics of creating games and animations in both applications, as well as be provided the opportunity to create their own game for possible entry into local competitions.
Prerequisite: Web Page Design

Instructional Strategies, Learning Practices, Activities, and Experiences

Content Introductions
Quizzes
Peer Review
Hands-On Practice Through Scratch Tutorials
Hands-On Practice Through Flash Tutorials
Independent Research
Final Projects
Guest Speakers/Success Stories
Competitions

Assessments

Inspirational/TED Talk Videos
Brainstorming Own Game (With Educational Tie-In)
Scratch Tutorials (11)
Scratch Reference Manual
Adobe Flash Tutorial
Flash Reference Manual
Scratch Final Project
(Progress Checks and Completed Project)
Adobe Flash Final Project
(Progress Checks and Completed Project)

Materials/Resources

Scratch Tutorials
Adobe Flash Tutorial
Microsoft® Office Software
iPad, VMware Horizon (Virtual Desktop) App

Adopted: 5/12/2012

Revised: 5/21/18

Scratch	
CONTENT/KEY CONCEPTS	**OBJECTIVES/STANDARDS**
Getting Started Exploring Animations Stories Games Diving Deeper <u>Related Vocabulary:</u> profile editor project page experimenting and iterating testing and debugging sequence sprite motion looks sound costume backdrop tips window remix reusing and remixing make a block backpack stage pass-it-on story pair programming Scratch Screening design demo abstracting and modularizing conditionals operators data variables and lists sensing	The students will be able to: • Be introduced to the concept of computational creation, in the context of Scratch. • Be able to imagine possibilities for their own Scratch-based computational creation. • Become familiar with resources that support their computational creation. • Prepare for creating Scratch projects by establishing Scratch accounts, exploring Scratch studios, creating design journals, and organizing critique groups. • Build on initial explorations of the Scratch environment by creating an interactive Scratch project. • Be introduced to a wider range of Scratch blocks. • Become familiar with the concept of sequence. • Practice experimenting and iterating while creating projects. • Be introduced to the computational thinking concepts of loops, events, and parallelism. • Become more familiar with the concepts of sequence. • Experiment with new blocks in the events, control, sound, and looks categories. • Explore various arts-themed Scratch programs. • Create an animated music video project. • Gain familiarity in and build understandings of the benefits of reusing and remixing while designing. • Develop greater fluency with computational concepts (events and parallelism) and practices (experimenting and iterating, testing and debugging, reusing and remixing). • Explore computational creation within the genre of stories by designing collaborative narratives. • Be introduced to the computational concepts of conditionals, operators, and data (variables and lists). • Become more familiar with the computational practices of experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing by building and extending a self-directed maze, pong, or scrolling game project. • Identify and understand common game mechanics. • Reflect on past experiences to self-assess current learning goals and needs. • Create a self-remix by extending a previously started project. • Be introduced to various hardware extensions that connect Scratch to the physical world. • Gain more fluency in computational concepts and practices by exploring the newest Scratch features (video sensing, cloning). • Experiment with designing learning experiences for others.

Scratch (Continued)	
CONTENT/KEY CONCEPTS	**OBJECTIVES/STANDARDS**
Related Vocabulary: (Continued) feedback fair arcade day puzzle jar brain dump video sensing cloning peer interviews hardware extension loops events parallelism control broadcast scripts presentation mode bitmap vector animation	15.4.12.A ~ Apply the creative and productive use of emerging technologies for educational and personal success. 15.4.12.B ~ Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship. 15.4.12.C ~ Develop criteria for analyzing hardware options to meet defined needs. 15.4.12.D ~ Evaluate emerging input technologies. 15.4.12.F ~ Compare and contrast network environments, including the function of network devices and connectivity issues. 15.4.12.G ~ Create an advanced digital project using sophisticated design and appropriate software/applications. 15.4.12.H ~ Use programming languages to develop logical thinking and problem solving skills. 15.4.12.I ~ Compare and contrast programming languages; select most appropriate one to complete a specific task. 15.4.12.L ~ Find and use primary documentation; employ an accepted protocol for citation. 15.4.12.M ~ Evaluate the impact of emerging technologies on various career paths and provide examples of industry certifications within the field.

Adobe Flash	
CONTENT/KEY CONCEPTS	**OBJECTIVES/STANDARDS**
Getting Started with Adobe Flash CS4 Drawing Objects in Adobe Flash Working With Symbols and Interactivity Creating Animations Creating Special Effects Preparing and Publishing Movies <u>Related Vocabulary:</u> flash stage timeline frames layers panels tools panel tools views colors options motion tweening motion path playhead user interface flowchart storyboard balance movement pixels normal mask masked folder guide symbol instances	The students will be able to: • Explain the Adobe Flash workspace while creating an Adobe Flash movie, application, or website. • Open a document and play a movie. • Create and save a movie. • Create, utilize, and manipulate a timeline. • Distribute an Adobe Flash movie. • Plan an application or a website. • Use the Flash drawing tools. • Select objects and apply colors. • Create, utilize, and manipulate drawn objects. • Utilize and manipulate text and text objects. • Create, utilize, and manipulate layers and objects. • Create symbols and instances. • Explore and utilize libraries. • Create buttons. • Assign actions to frames and buttons. • Import graphics. • Create motion tween animations. • Create classic tween animations. • Create frame-by-frame animations. • Create shape tween animations. • Create movie clips. • Animate text. • Create a mask effect. • Add sound. • Add video. • Create an animated navigation bar. • Create character animations using inverse kinematics. • Create 3D effects. • Publish movies. • Reduce file size to optimize a movie. • Create a preloader. • Use HyperText Markup Language (HTML) publishing settings.

Adobe Flash (Continued)	
CONTENT/KEY CONCEPTS	**OBJECTIVES/STANDARDS**
Related Vocabulary: (Continued) up-button over-button down-button hit-button release key press roll over drag over bitmap image vector graphics motion guide shape tweening morphing movie clip symbol mask layer embedded video progressive downloading steaming video inverse kinematics (IK) bandwidth profiler preloader template dimensions playback quality window mode html alignment scale flash alignment	15.4.12.A ~ Apply the creative and productive use of emerging technologies for educational and personal success. 15.4.12.B ~ Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship. 15.4.12.C ~ Develop criteria for analyzing hardware options to meet defined needs. 15.4.12.D ~ Evaluate emerging input technologies. 15.4.12.F ~ Compare and contrast network environments, including the function of network devices and connectivity issues. 15.4.12.G ~ Create an advanced digital project using sophisticated design and appropriate software/applications. 15.4.12.H ~ Use programming languages to develop logical thinking and problem solving skills. 15.4.12.I ~ Compare and contrast programming languages; select most appropriate one to complete a specific task. 15.4.12.L ~ Find and use primary documentation; employ an accepted protocol for citation. 15.4.12.M ~ Evaluate the impact of emerging technologies on various career paths and provide examples of industry certifications within the field.