CodeHS

Utah Creative Coding Syllabus

1 semester for Middle School (85 contact hours)

Course Overview and Goals

Creative Coding introduces students to game design and programming. Students learn to design, develop, and publish games. Throughout the course, students learn about major programming topics like variables, functions, conditionals, and mouse/keyboard events while building fun and interactive games that they can play and share with others. Students practice computational thinking skills as they plan, develop, and test their programs.

Students build three different games in the last three modules in this course. As they follow tutorial-like activities, they get to practice their computational thinking and programming skills, learning new concepts along the way. By the end of these projects, they'll have fully functional and interactive games that they can customize and share with their friends.

Note: it is recommended that students complete the projects in the order listed, since topics build off each other, but not all projects must be completed if time doesn't permit. Developing 1 or 2 of the projects will still hit all of the Utah Creative Coding standards.

Learning Environment

The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons contain a variety of activities, from video tutorials, example programs to explore, interactive web pages, and hands-on programming exercises.

Programming Environment

Students write and run JavaScript programs in the browser using the CodeHS editor.

Technology Requirements

To complete all activities and exercises in this course, students must have access to the 3rd party sites and tools listed here: Creative Coding Course Links

Prerequisites

There are no prerequisites for this course, though prior exposure to the basics of programming would always help. This course is meant to be a first-time introduction to computer science and does not require students to come in with any computer programming experience. Students should be familiar with basic geometry and the Cartesian (x, y) coordinate system to represent points in a plane.

More Information

Browse the course content: https://codehs.com/course/20516

Course Breakdown

Module 1: Intro to Game Design (1 weeks / 3-5 hours)

Students reflect on their game-playing experience and learn more about major video game topics and categories.

Objectives / Topics Covered	● Introduction to Games ● Gameplay Elements ● Game Perspectives ● Game Genres
Example Assignments / Labs	● Introduction to Games ○ Students begin to reflect on the variety of games they have played and what actually makes a game a game. ● Unpacking a Game ○ Students explore and learn to identify game elements, mechanics, and components that make up the overall gameplay experience. They play a free online game in order to practice noticing the different pieces involved. ● Categorizing Games ○ Students learn about game perspectives and genres as a way to categorize games by their styles and gameplay elements. They practice by classifying a few popular games.

Module 2: Programming in JavaScript (2 weeks / 6-8 hours)

Students learn the basics of JavaScript including variables, user input, mathematics, and functions.

Objectives / Topics Covered	● Computational Thinking ● Intro to JavaScript ● Printing ● Variables ● User Input ● Basic Math ● Basic Functions ● Pseudocode
Example Assignments / Labs	● How to Think Like a Programmer ○ Students learn about computational thinking techniques and practice decomposing processes and writing repeatable algorithms. ● Intro to JavaScript ○ Students encounter JavaScript for the first time and learn how to print text to the console. ● Variables ○ Students practice using variables in a variety of contexts – making

	purchases, olympic winners, and a hiking app. ● User Input ○ Students develop a program that asks users for input to complete a Mad Lib story. ● Basic Math ○ Students use mathematical calculations to write a program for a bookstore and animate a graphical pigeon. ● Basic Functions ○ Students apply their knowledge of calling and defining functions to create a custom function of their own!

Module 3: Getting to Know Libraries (2 weeks / 6-10 hours)

Students learn about the JavaScript game design library and how to use it to create the basic structure of an animated or interactive program.

Objectives / Topics Covered	● Introduction to Libraries ● Program Structure ● Canvas ● Sprites and Properties
Example Assignments / Labs	● Introducing Libraries ○ Students get their first glimpse of how powerful, and fun, the JavaScript Game library is. ● Program Structure ○ Students create two animations using setup() and draw() functions. ● Understanding the Canvas ○ Students create their first canvas and practice using the coordinate system to place boxes on shelves. ● Your First Sprites ○ Students show their canvas and sprite expertise by recreating a classic Mario screenshot with basic shapes. ● The Physics of Sprites ○ Students work through a variety of simple programs in order to better understand all of the physics-related sprite properties the library offers.

Module 4: Project: Mini Golf (4-5 weeks / 20-25 hours)

Students incrementally develop a mini golf game, learning more about important computer science topics like booleans and if/else statements.

Objectives / Topics Covered	● Project Design Process ● Sprite and World Configuration ● Booleans ● If/Else Statements ● Player Interactions ● Adding Effects ● User Interfaces
Example Assignments	● As students progress through this project module's lessons, they develop an

/ Labs	interactive mini golf game. They practice working with sprites and their properties, and after learning about Booleans and if/else statements, enable the player to interact with the sprites. At the end of the project, students get to be creative and make a 2nd mini golf hole of their own design!

Module 5: Project: Projectiles (4-5 weeks / 20-25 hours)

Students incrementally develop a projectiles game, learning about Groups and Tiles, while improving their fundamental program development skills.

Objectives / Topics Covered	● Project Design Process ● Sprite and World Configuration ● Images ● Groups ● Tiles ● Booleans ● If/Else Statements ● Player Interactions ● User Interfaces
Example Assignments / Labs	● In this project module, students develop an Angry Bird's-like projectiles game. They get more practice with previous skills, like sprite development and conditional statements, while learning about Groups and Tiles. At the end of the project, students can showcase their creativity by designing and building a second level of projectiles of their choice.

Module 6: Project: Jumper (4-5 weeks / 20-25 hours)

Students incrementally develop a platform jumper game, learning how to animate game components with sprite sheets, while improving their fundamental program development skills.

Objectives / Topics Covered	● Practice programming with topics introduced in previous projects ● Sprite sheets ● Sprite animation ● Camera movement ● Player mechanics
Example Assignments / Labs	● Students develop a platform jumper game in this project. In doing so, they get more practice with Groups and Tiles, and learn how to use sprite sheets to animate their game's components.