SCHOOL DISTRICT OF THE CHATHAMS
CURRICULUM PROFILE

CONTENT AREA(S): iSTEM **GRADE LEVEL(S):** 8th Grade

COURSE: Digital Game Design **TIME FRAME:** Quarterly (39-40 days)

I. Course Overview

This course will allow students to explore the basics of computer hardware components and develop computational thinking skills, which will assist students in problem solving through creating game applications in Scratch. Students will follow the Engineering Design process to create aspects of game design such as logic mapping, intuitive user interactions, coding, developer testing, debugging, peer critiquing, user testing, and user feedback. Students will gain proficiency in computational thinking principles while implementing "if else" statements, loops, functions, variables, and operators. The Digital Game Design course will utilize 21st-Century learning skills such as collaboration, communication, and teamwork to create solutions to real world problems.

II. Units of Study

1.) Introduction to Coding in Scratch *(~20 days)*
2.) Advanced Coding in Scratch *(~20 days)*

III. Essential Questions

Unit 1: Introduction to Coding in Scratch

- How has the evolution of computer technology progressed and identify where you see the future of computer technology going?
- Why did certain technologies originally fail or became more popular when re-introduced by another company?
- How can a thorough understanding of the nature of a problem, attention to detail, and an understanding of criteria and constraints impact the development and execution of programming code?
- How can coding be applied to simplify a task or automate, process, or solve a problem?
- How can a better understanding of programming logic affect the way a program is written or executed?
- How can certain programming blocks be used to help condense the amount of code we have to right?
- How can the application of operators and variables help simplify and organize a developer's code?

Unit 2: Advanced Coding in Scratch

- How can the application of the Engineering Design Process help us better develop our projects?
- How can the application of functions help simplify and organize a developer's code?
- How can we create a program that keeps a player engaged in the game?

IV. Learning Objectives

NJSLS - Design Technology

- 8.2.8.A.1 Research a product that was designed for a specific demand and identify how the product has changed to meet new demands (i.e. telephone for communication - smart phone for mobility needs).
- 8.2.8.B.5 Identify new technologies resulting from the demands, values, and interests of individuals, businesses, industries and societies.
- 8.2.8.C.3 Evaluate the function, value, and aesthetics of a technological product or system, from the perspective of the user and the producer.
- 8.2.8.C.4 Identify the steps in the design process that would be used to solve a designated problem. (Design Process)
- 8.2.8.C.6 Collaborate to examine a malfunctioning system and identify the step-by-step process used to troubleshoot, evaluate and test options to repair the product, presenting the better solution.
- 8.2.8.C.8 Develop a proposal for a chosen solution that include models (physical, graphical or mathematical) to communicate the solution to peers.
- 8.2.8.D.1 Design and create a product that addresses a real-world problem using the design process and working with specific criteria and constraints.
- 8.2.8.D.2 Identify the design constraints and trade-offs involved in designing a prototype (e.g., how the prototype might fail and how it might be improved) by completing a design problem and reporting results in a multimedia presentation, design portfolio or engineering notebook.
- 8.2.8.D.3 Build a prototype that meets a STEM-based design challenge using science, engineering, and math principles that validate a solution.
- 8.2.8.E.2 Demonstrate an understanding of the relationship between hardware and software.
- 8.2.8.E.3 Develop an algorithm to solve an assigned problem using a specified set of commands.
- 8.2.8.E.4 Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms).

Technology Integration | NJSLS 8.1

- 8.1.8.A.1 Demonstrate knowledge of a real world problem using digital tools.
- 8.1.8.A.3 Use and/or develop a simulation that provides an environment to solve a real world problem or theory.

21st Century Integration | NJSLS 9

- 9.2.8.B.3 Evaluate communication, collaboration, and leadership skills that can be developed through school, home, work, and extracurricular activities for use in a career
- 9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

Career Ready Practices

- CRP2. Apply appropriate academic and technical skills.
- CRP4. Communicate clearly and effectively and with reason.
- CRP6. Demonstrate creativity and innovation.

- CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
- CRP11. Use technology to enhance productivity.
- CRP12. Work productively in teams while using cultural global competence.

Interdisciplinary Connections

NGSS | Science

- MS-ETS1-2. Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.
- MS-ETS1-3. Analyze data from tests to determine similarities and differences among several design solutions to identify
  the best characteristics of each that can be combined into a new solution to better meet the criteria for success.
- MS-ETS1-4. Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process
  such that an optimal design can be achieved.

V. Instructional Materials

Core Materials:

- Coding in Scratch: Projects Workbook [DK Workbook]
- Computer Coding: An Introduction to Computer Programming [DK Workbook]
- Coding in Scratch: Games Workbook [DK Workbook]
- Scratch Challenge Workbook [DK Workbook]
- Coding Games in Scratch: A Step-by-Step Visual Guide to Building Your Own Computer Games by Jon Woodcock
- How to Code in 10 Easy Lessons: Learn How To Design And Code Your Very Own Computer Game by Sean McManus
- Code Your Own Games!: 20 Games to Create with Scratch by Max Wainewright
- G-Suite for Education
- Scratch Programming Software
- Arduino Hardware & Software
- NAO Robots
- Choreograph Programming Software
- Teacher computer with Internet access and projector/Smart Board
- Document Camera
- Desktop Computing Devices

Supplemental/District Created Materials:

- Class Introduction Presentation
- Scratch Permission Slip
- Old Computers Video
- Scratch Intro Activities (Google Classroom)
- Animation Project Rubric / Handout
- Quiz Game Show Rubric / Handout
- Amazing Maze Rubric / Handout
- Hungry Caterpillar Rubric / Handout
- Maze Notes
- Apple Clicker Game Demo

- Variables Notes
- Quiz Game Demo
- Coding Sample Notes
- Animation Notes
- Design Process Handout
- Design Process and Functions Presentation
- Functions Notes on Scratch
- Elements of a Good Game Activity (Google Classroom)
- Elements of a Good Game Presentation
- Classic Arcade Game Group Checklist
- Classic Arcade Game Design Process
- Classic Arcade Game Rubric / Handout
- Arduino Rubric / Handout
- Create your own Rubric / Handout
- Final Project Design Process
- Design your own Scratch Game Rubric / Handout
- Create your own NAO Game Design Process
- Design you own NAO Game Rubric / Handout
- Simulate Gravity Notes
- Launching Objects Notes
- Side Scroller Notes

VI. Key Performance and Benchmark Tasks

Assessment Methods:

- Students will complete approximately five (5) projects between unit 1 and 2.
- When a student completes a project, s/he will complete a checklist, reflect on their work and answer thoughtful questions on their design process.
- A rubric is outlined on the checklist sheet, delineating the project parameters and expectations.

Summative:

Unit 1: Introduction to Coding in Scratch

- Students will be able to create a program that allows Sprites (characters) to move across the screen and interact together in varying circumstances.
- Students will be able to import and design their own Sprites (characters) that show movement using multiple costumes.
- Students will be able to utilize the mathematical coordinate plane in Scratch to choose precise movement points for Sprites (characters).
- Students will be able to create a program that runs autonomously in Scratch.
- Students will be able to create an interactive game in Scratch.
- Students will be able to incorporate a data variable into an interactive program and manipulate that data variable to provide meaningful outputs when appropriate.
- Students will be able to incorporate a broadcast in a logical manner.
- Students will be able to incorporate 'If Else' and 'Forever' statements when necessary in a program.
- An Interactive Commercial / Animation
    - Animation Graded Rubric

- Quiz Game Show
    - Quiz Game Show Graded Rubric
- Amazing Maze / Hungry Caterpillar
    - Amazing Maze Graded Rubric
    - Hungry Caterpillar Graded Rubric

Unit 2: Advanced Coding in Scratch

- Students will be able to demonstrate that functions, data variables, and broadcasts can be used in a logical manner.
- Create complex programs with multiple working parts.
- Demonstrate the use of the Engineering Design Process.
- Incorporate elements of a good game into a project in a logical manner.
- Troubleshoot complex programming issues.
- Brainstorm a final project that meets various criteria and constraints.
- Create a complex game which utilizes functions, data variables, and broadcasts in a logical manner.
- Create a game that incorporates elements of a good game.
- Create various interactive screens within a game.
- Propose a final project that meets various criteria and project constraints.
- Create a complete game which has a definitive beginning and end.
- Classic Arcade Game
    - Classic Arcade Game Graded Rubric
- Final Project
    - Design Your Own Scratch Game Graded Rubric
    - Programming with NAO Graded Rubric
    - Arduino Graded Rubric

<u>Formative:</u>

- Peer Feedback: TAG (Tell, Ask, Give) Sticky Notes
- Peer Feedback Form
- Self-Reflection: 2 Stars & 1 Wish
- Critique Guide
- Reflective Exit Tickets/Slips
- An Interactive Commercial / Animation
    - Animation Checklist
- Quiz Game Show
    - Quiz Game Show Checklist
- Amazing Maze / Hungry Caterpillar
    - Amazing Maze Checklist
    - Hungry Caterpillar Checklist
- Classic Arcade Game
    - Classic Arcade Game Checklist
- Final Project
    - Design Your Own Scratch Game Checklist
    - Programming with NAO Checklist
    - Arduino Checklist

Alternative:

- Student choice is built into each project, which makes each project unique for each and every student.
- Adjustments to assessment criteria and assessments themselves are described below in Section VII.

VII. Accommodations & Modifications for Special Education, Students at Risk for School Failure, English Language Learners, Gifted & Talented, and 504s

Special Education

- Student choice in projects to allow for appropriate skill levels to be applied.
- Clarify and repetition of expectations, review of expectations at the start of class, highlighting expectations on student hardcopies, provide specific tasks as needed to clarify goals.
- Support of student focus: verbal prompts, visual cues (lights out, etc.).
- Positive reinforcement.
- Pacing and guidance in long term projects.
    - Work chunked out based on tasks, individual check ins.
    - Extended projects are broken down into manageable tasks with frequent check-ins from the teacher.
- Provision of Written Notes and Directions:
    - Maze Notes
    - Variables Notes
    - Quiz Game Demo
    - Coding Sample Notes
    - Animation Notes
    - Design Process and Functions Notes
    - Functions Notes on Scratch
    - Elements of a Good Game Notes
    - Final Project Design Process Guide
    - Create Your ONAO Game Design Process Guide
    - Classic Arcade Game Group Checklist
    - Simulate Gravity Notes
    - Launching Objects Notes
    - Side Scroller Notes

English Language Learners

- Use of Google Translate to assist students with instructions and lessons so they can follow along.
- Adjust goals to allow for language acquisition.
- Visual prompts and demonstrations.
- Teacher modeling of skills.

- Simplified written and verbal instructions. Include written instructions to supplement verbal.
- Preferential seating.

Gifted & Talented

- Access to additional materials to develop ideas and project details.
    - Extension Activities Folder
    - In class Scratch Activity Books
- Student leadership opportunities.

Students at Risk of School Failure

- Student choice in projects to allow for appropriate skill levels to be applied.
- Clarify and repetition of expectations, review of expectations at the start of class, highlighting expectations on student hardcopies, provide specific tasks as needed to clarify goals.
- Support of student focus: verbal prompts, visual cues (lights out, etc.).
- Positive reinforcement.
- Pacing and guidance in long term projects: Work chunked out based on tasks, individual check ins.
- Extended projects are broken down into manageable tasks with frequent check-ins from the teacher.

504s

- Completely dependent on the student's 504 plan.
    - If the student cannot utilize computers or look at screens, research, planning, and computer-based learning experiences can be done on paper.
    - If the students' level of mobility is limited, making it difficult for the students to navigate the classroom, the student will be assigned a buddy to help with acquiring the necessary materials and supplies.
    - If the students' fine or gross motor skills are impacted, s/he will receive assistance from the teacher for the specific artistic skills that require them.

GENERAL NOTES:

- The order in which the units are taught can be adjusted at the teacher's discretion.
- Days are fluid and some activities may extend longer.
- Lessons and units will be adjusted as per students' prior knowledge.