

MakeCode Arcade Coding

Total Number of Lessons: 5 (5th & 6th grade)

Total Estimated Time for Module: 225-250 minutes

SECTION 1: MODULE OVERVIEW AND CURRICULAR CONNECTIONS – REQUIRED

- **Module goal:** Students will be able to use the MakeCode Arcade programming environment to design, code, test, and improve an original video game for the BrainPad Arcade using block-based programming.
- **Connection to standards:**
 - **3-5.PA.2** Use digital tools to gather, manipulate, and modify data for use by a program.
 - **3-5.PA.3** Implement problem solutions using a block-based visual programming language.
 - **6-8.PA.2** Implement problem solutions using a programming language that includes looping behavior, conditional statements, logic, expressions, variables, and functions.
 - **6-8.PA.3** Demonstrate dispositions amenable to open-ended problem solving and programming (e.g., comfort with complexity, persistence, brainstorming, adaptability, patience, propensity to tinker, creativity, accepting challenge).
- **RET materials/ideas to be leveraged:** I am going to use the information I learned from Computational Thinking and Scratch with Mr. Dingler.

SECTION 2: OVERVIEW OF MODULE FRAMEWORK – REQUIRED

- **Real-world context:** Coding is used in many careers beyond video games, including robotics, engineering, medicine, education, and app development. Learning to program helps students develop problem-solving and computational thinking skills that are valuable in today's technology-driven world.
 - **Possible Lesson Ideas:** To introduce this module, I can have students brainstorm all the technology they interact with in a typical day (phones, games, streaming services, cars, traffic lights, robots, etc.). Discuss how each relies on programming and how coding is used in careers like engineering, medicine, agriculture, education, and game design. Then I can have students watch a simple video game and ask students "What instructions did the programmer have to write to make this happen?" This could be a great unplugged lesson using arrow cards

or words to make a game piece get out of a maze before even using MakeCode Arcade and BrainPads.

- **Background STEM content:** Students will build foundational computer science skills using MakeCode Arcade. They will learn programming concepts such as sequencing, events, loops, conditionals, variables, and debugging while developing computational thinking and problem-solving skills. Through designing, testing, and refining their own games, students will experience the iterative process used by software developers and engineers to create technology.
 - **Possible Lesson Ideas:** Students complete one or two guided MakeCode Arcade tutorials to learn how to create sprites, add movement, use controller buttons, and test their programs. Then, students will go back into their tutorial and make some changes to make it their own!

- **Description of Main Activity/Project:** Students will begin by exploring the MakeCode Arcade platform through a series of guided tutorials to learn the fundamentals of block-based programming, including creating sprites, using controller inputs, adding events, and debugging code. Once they have developed these foundational skills, students will apply their knowledge to design, code, test, and refine an original video game for the BrainPad Arcade. Throughout the project, students will use computational thinking and the engineering design process to solve problems, improve their programs through iteration, and create a fun, interactive game that demonstrates their understanding of computer science concepts.
 - **Possible Activity/Project Ideas:**
 - Students will begin by completing guided MakeCode Arcade tutorials to learn the coding platform and practice programming concepts such as sprites, controller inputs, events, loops, conditionals, variables, and debugging. They will then brainstorm, plan, and create an original video game for the BrainPad Arcade, testing and improving their game through multiple iterations before sharing it with classmates.

The project will take approximately 4-5 class periods:

- Days 1–2: Introduction to coding and real-world applications; complete MakeCode Arcade tutorials.
- Day 3: Continue tutorials and begin planning an original game.
- Day 4: Design, code, test, debug, and improve the game.
- Day 5: Final testing, game showcase, peer feedback, and reflection.

SECTION 3: MODULE SEQUENCING AND ASSESSMENT – REQUIRED

- **Brief description of sequenced learning objectives:** Students will begin by exploring how coding is used in everyday life and learning the basics of the MakeCode Arcade programming environment through guided tutorials. As they progress, they will develop foundational programming skills, including creating sprites, using events, controller inputs, loops, conditionals, variables, and debugging. Students will then apply these skills to plan, design, and program an original game for the BrainPad Arcade. Throughout the module, they will test, refine, and improve their games using computational thinking and the engineering design process, culminating in a completed game that demonstrates their understanding of key computer science concepts.
- **Brief description of formative and summative assessment approaches:** Student progress will be assessed throughout the module through teacher observations, guided questioning, completion of MakeCode Arcade tutorials, coding checkpoints, debugging activities, and peer feedback. At the end of the module, students will demonstrate their learning by creating an original video game for the BrainPad Arcade, which will be evaluated using a rubric based on programming skills, game functionality, creativity, and problem-solving. Students will also complete a brief reflection on their design process and coding experience.
- **Recommendations for implementation:** Introduce MakeCode Arcade through guided tutorials before asking students to create their own games, as this builds confidence with the coding platform and programming concepts. Encourage students to frequently test and debug their code rather than waiting until the end of the project. Provide planning time for students to sketch their game ideas before coding, and allow opportunities for peer collaboration and feedback throughout the design process. Ensure each student has access to a device with MakeCode Arcade and a BrainPad Arcade for testing, and build in extra time for troubleshooting hardware or coding issues. Remind students to save their work often to prevent losing progress.
 - I am also wanting a subscription to Chat GPT Pro as an extra instructional resource to support students during the BrainPad Arcade project. I can use it to help students troubleshoot coding errors, explain programming concepts, brainstorm game ideas, and develop additional coding challenges. It would also allow me to provide more individualized support and extensions for students with different skill levels. While it is not required for the project, it would be a valuable tool to enhance student problem-solving, creativity, and confidence as they learn to code.

SECTION 4: MODULE FILES – ENCOURAGED/OPTIONAL

● **List of attached files:**

- o Makecode Arcade - this is the platform where students will design and make their code to transfer to a BrainPad: <https://arcade.makecode.com/>
- o Makecode Tutorial - this is a great tool to make my own tutorials for students if the ones on the website are too hard/confusing. <https://makecode.com/tutorial-tool>
- o Chat GPT Pro - <https://chatgpt.com/>