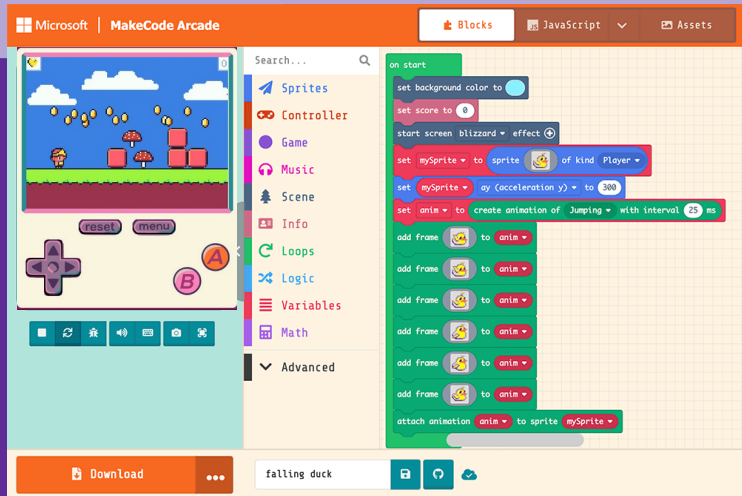


MakeCode Arcade Coding

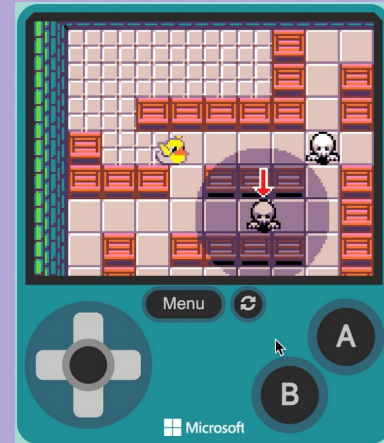
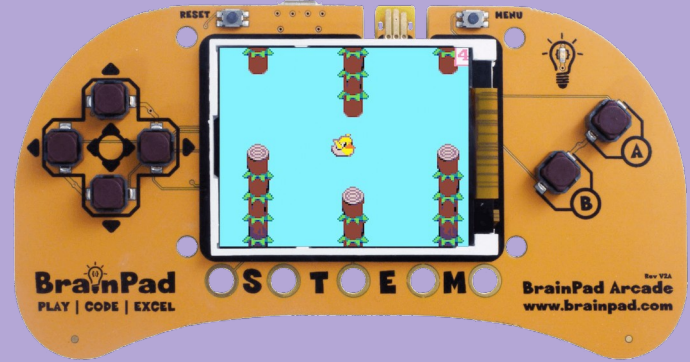
Allison Moore

PLTW - Liberty Elementary School



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.



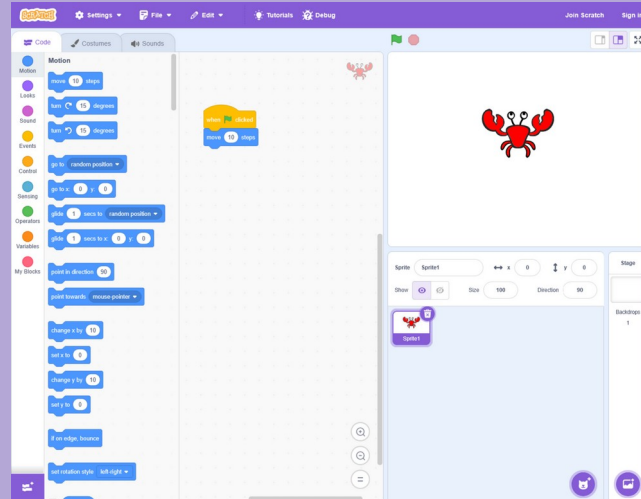
Connection to Standards



- **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 in the Classroom

I am going to use the information I learned from Computational Thinking and Scratch with Mr. Dingler to create lessons using MakeCode Arcade, which is similar to Scratch.



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.



Background STEM Content

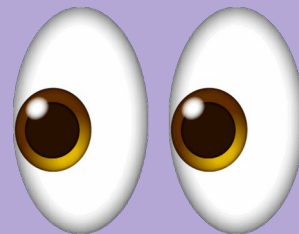


Students will build foundational computer science skills using MakeCode Arcade. They will learn programming concepts such as sequencing, events, loops, conditionals, variables, and debugging while developing computational thinking and problem-solving skills. Through designing, testing, and refining their own games, students will experience the iterative process used by software developers and engineers to create technology.



Main Activity

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.



A Closer Look

- 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.

Set the background color



Open the
Scene
 toolbox drawer and drag
set background color
 []
 into the empty on start
 container already in your
 workspace.

▶ What does that mean?

💡 Feel free to choose your own color if you don't like the swatch in the block.

Click the button that says **Next**
 to go to the next step of the

Toolbox

Search...

Sprites

Controller

Scene

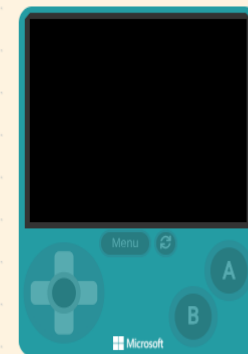
Info

Loops

Variables

Math

on start

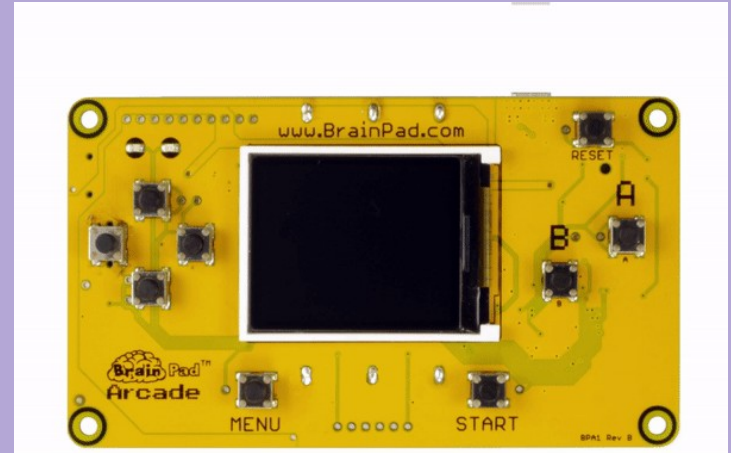


Download



Assessments

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

I will 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. I will encourage students to frequently test and debug their code rather than waiting until the end of the project. I will provide planning time for students to sketch their game ideas before coding, allow opportunities for peer collaboration and feedback throughout the design process, and build in extra time for troubleshooting hardware or coding issues. And finally I will remind students to save their work often to prevent losing progress.



Files/Documents

- Makecode Arcade - this is the platform where students will design and make their code to transfer to a BrainPad: <https://arcade.makecode.com/>
- 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>
- Chat GPT - <https://chatgpt.com/>

