Project: Build a Video Game

(Also available in WeScheme)

Students build a video game of their own design! Every game must have a player (their avatar), a danger (something to avoid), and a target (something to chase). Students build their world using function composition, animate characters through linear functions, handle keypresses with piecewise functions, detect boundaries with compound inequalities, and detect collisions with the distance formula.

Lesson Goals	Students will be able to… • recognize meaningful applications of multiple algebraic concepts. • express their creativity through the design of a game context. • share their completed video game with friends and family to play.
Student-facing Lesson Goals	• Let's create a video game!
Prerequisites	• Simple Data Types • Contracts • Simple Inequalities • Functions: Contracts, Examples & Definitions • Solving Word Problems with the Design Recipe • Problem Decomposition • Piecewise Functions and Conditionals • Collision Detection - Distance and Inequality
Materials	• PDF of all Handouts and Page • Blank Game Starter File • Google Draw template (Google) • Lesson Slides • Printable Lesson Plan (a PDF of this web page)

Preparation	• Following the default lesson order for Bootstrap:Algebra (e.g., teaching every lesson) will result in students building video games. That said, some teachers prefer to save all *game-focused lessons* for the end of the year, allowing students to create games in 1-2 weeks. • Decide if you want to teach concepts first - and culminate with students creating their games… or take an integrated approach, where students gradually add to their games over a longer period of time. It is up to you!
Key Points for the Facilitator	• Students add to their game's code during a variety of lessons in Bootstrap:Algebra. • This project can be extended. Students can add their own "cheat codes" (piecewise/conditional functions). They can also add more complex movement (using quadratic, exponential, and trigonometric functions).

Build Your Own Video Game *flexible*

Overview

This project can be used as the capstone for Bootstrap:Algebra, or students can build their video games intermittently over a longer period of time. The video game project is designed to provide real-world and engaging connections to the following mathematical concepts: Coordinates; Simple and Compound Inequalities; Domain and Range; Composing Functions; Connecting Representations and Defining Functions; Rotation, Dilation, and Translation; the Distance Formula and Pythagorean Theorem; Piecewise Functions.

Launch

There are six phases of video game creation. Each phase (except for the first and the last) requires that you teach:

- up to four prerequisite lessons during which students learn relevant coding skills and algebra concepts
- one lesson where students use what they've learned to update the code of their Game Starter File

In the first two phases (below), students conceptualize their game and then add images to their Game Starter File. The table below highlights (1) the lessons you will teach to reinforce relevant algebra and coding concepts, and (2) the lessons where students will update their personal video games.

	Phase	Prerequisites	Building the Game
0	Brainstorming	*NA*	Coordinates and Game Design
1	Game Imagery	Order of Operations, Simple Data Types, Contracts, Defining Values	Making Game Images

By the end of Phase 0, each student will have created a "screenshot" of their video game by inserting images that they find on the web into a Google Draw template (Google). Along the way, students will think about horizontal and vertical position as represented by ordered pairs.

In Phase 1, students acquire the skillset needed to actually add game images to section 0 of their Game Starter File. The programming concepts that students learn here form a foundation upon which to build, and will prove essential as students work through the remaining lessons. Students actually edit and update their games during Making Game Images.

During Making Game Images, we recommend printing and distributing the student-facing rubric. This document will highlight the scope of the project - and your expectations - for students. Direct your students' attention to Rubric Row 1 ("Images") to clarify your requirements for this component of their video game. (We encourage you, of course, to edit the rubric to meet your classroom needs!)

Investigate

In the remaining phases, students add functionality to their games, drawing on more advanced coding and algebra concepts along the way.

#	Phase	Prerequisites	Building the Game
2	Character Movement	Functions: Contracts, Examples & Definitions, Solving Word Problems with the Design Recipe	Functions for Character Animation
3	Boundaries	Problem Decomposition, Simple Inequalities, Compound Inequalities: Solutions & Non-Solutions	Sam the Butterfly - Applying Inequalities
4	Player Movement	Piecewise Functions and Conditionals	Player Animation
5	Collisions	*NA*	Collision Detection - Distance and Inequality

We encourage you to point out the relevant rubric rows as you progress through the above series of lessons.

In Bootstrap:Algebra, we offer many additional lessons where students can develop their math and coding skills. *The lessons outlined above are the ones considered **essential** for video game creation.* We encourage you to integrate additional lessons that meet your students' needs.

Synthesize

- Encourage students to self-assess and revise their work as they progress through the lessons. Peer review is a powerful tool if time allows.

- Celebrate students' work! Many Bootstrap teachers arrange **video game launch parties**, essentially creating a "science fair for the math department", complete with tri-fold posters explaining elements of the game and the math at work behind the scenes.