The Numbers Inside Video Games

(Also available for Pyret)

Students reverse engineer a video game and research what takes to create a video game.

Prerequisites	None
Relevant Standards CC-Math	*Select one or more standards from the menu on the left (⌘-click on Mac, Ctrl-click elsewhere).*
Lesson Goals	Students will be able to: • Identify the objects in a video game that are changing. • Use math language to describe what is changing about each object. • Understand the time, money, and resources it takes to create a popular video game.
Student-Facing Lesson Goals	• I can identify the objects in a video game. • I can use math vocabulary to describe what is changing about each object. • I understand the time, money, and resources it takes to create a popular video game.
Materials	• Lesson slides template (Google Slides) • Reverse Engineer worksheet (HTML (Page 3), Google Doc) • NinjaCat demo game (WeScheme)
Preparation	• Make sure all materials have been gathered • Decide how students will be grouped in pairs
Key Points for the Facilitator	• Students will need their own Google accounts. • Take care to manage student expectations about what their game will be like. Modern games are very complex!
Supplemental Resources	• Coordinates (Quizizz) • The Awesome Coordinate Plane Activity (Desmos Activity) • Boat Coordinate Game (Geogebra) • Coordinate Grid Exploration (Geogebra)

For a textbook-like version of materials similar to these, you may wish to see the prior unit-based version.

Reverse Engineering a Video Game

25 minutes

Overview

Students play a simple video game, and gradually break it down into parts. Doing so reveals how coordinates play a crucial role in video games, and how animation is created via equations that govern the changing values of those coordinates.

Launch

Play the NinjaCat demo game onscreen while students watch. Purposely make mistakes while playing the game, which should elicit responses/direction from students.

Pedagogy Note!

This pedagogy has a rich grounding in literature, and is used throughout this course. In the "Notice" phase, students are asked to crowd-source their observations. No observation is too small or too silly! By listening to other students' observations, students may find themselves taking a closer look at the game. The "Wonder" phase involves students raising questions, but they must also explain the context for those questions. Sharon Hessney (moderator for the NYTimes excellent What's going on in this Graph? activity) sometimes calls this "what do you wonder…and **why**?". Both of these phases should be done in groups or as a whole class, with adequate time given to each.

Take turns playing the game in pairs. After you've both had a chance to play, write down what you *notice* about the game on Notice and Wonder (Page 2). "Notice"s should be statements, not questions - What stood out to you? What do you remember?

Crowdsource students' Notices.

What do you *wonder* about the game? What questions do you have about how it works? Write these down on Notice and Wonder (Page 2).

Crowdsource students' Notices.

Investigate

Students complete the Reverse Engineer a Video Game (Page 3) worksheet in pairs.

- 1st Column: What are all the various *things* in this game? *(A dog, Clouds, etc.)*
- 2nd Column: For each of those "things", what is changing about them? *(Location, Position, etc.)*
- 3rd Column: For each change, how is it modeled mathematically? *(x-coordinate, y-coordinate, amount, etc.)*

Possible Misconceptions

- Students are likely to describe what the character is *doing*, as opposed to *what changes*. For example: "The dog is moving to the left" is not actualy describing the property being changed (position, place, location, etc).
- Students may write down what they *hope* is changeable, as opposed to what actually changes. It's common for students to say they cat's costume is changing, because they assume the cat will somehow "level up" if they get enough points.

Synthesize

The main idea here is to understand that while we see images on a screen, the computer only sees a small set of numbers, which uniquely model the state of the game. The way those numbers change determines how the game behaves, and we can add features to the game if we're willing to keep track of more numbers.

- If the x- and y-coordinates are each numbers, how many numbers does it take to represent a single frame of the video game?
- How are those numbers changing - or *varying* - as the game plays? When do they increase? Decrease?
- How many numbers would we need if the dog could also move up and down?
- How many numbers would we need to have a two-player game?
- How many numbers would we need if the entire game was in 3d?
- How many numbers would we need to make a modern game?

Connecting to Real Games 25 minutes

Overview

Students apply this way of thinking to more complex, real-world games. They also get a sense for how much work is involved in creating games like that.

Launch

Ask students to share out their favorite current video game. Write the names of the games on the board.

Investigate

Let students choose a current, popular game to discuss.

Collect students estimates for each of the questions below.

- How long do you think it took to create that game?
- How *many people* do you think it takes to create a game like that?
- How *much money* does it take to create a game like that?

Optional: Once students have made their estimates, have students use the Internet to research these questions and compare the actual numbers to their estimates.

The goal here is not to discourage students from the possibility of eventually creating a game like their favorite game, but to manage expectations given our limited resources (time, money, and people). By starting with this game project, students are learning transferable skills that can help them later on in learning new programming languages and creating bigger projects.

Synthesize

- What does this tell us about making modern games?
- Are we likely to create games like the ones you researched?

The 3d, two-player version of NinjaCat needed a lot more numbers than the simple one you saw here, *but the actual concepts at work are the same* . Even if we don't have time to make games like the ones we chose here, you'll learn the same concepts just by making a simpler one.

Closing

- Share-back: have students share their estimates with the class. Was anything drastically higher or lower than they expected?