Code, Play, Create

Activity URL: https://codecombat.com/play/hoc-2018

Overview

This activity includes ten different game levels designed to enable students to design their very own arcade game in just an hour. Each of the first nine levels contains some starter code. Students will finish the code, then play the game to beat the level. In the final level, students will creatively apply what they've learned to design their own game from scratch.

This lesson plan will help you orient students to the basics of programming before they begin the game.

Level: Beginner

Time: One 60-minute session

Materials:

- *Computer for each student or pair*
- *Printable Syntax Guide for each student or pair:*
  - *Python Guide*
  - *JavaScript Guide*

Objectives

- Understand that everything a computer does is the result of a program written in code by a programmer.
- Understand that changes to a program result in changes to the computer's behavior.
- Understand the roles of objects, properties, coordinates, and event handlers in game development.
- Use code to make creative products for self-expression.
- Modify, remix, and incorporate portions of an existing program to develop something new.

CSTA Standards

- **1B-AP-12** Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
- **1B-AP-15** Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
- **3A-AP-16** Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

Opening Activity (10 mins)

Interact

Have students tell a partner about a video game or mobile app game they like to play, then share out with the class. What do they like about that game? What makes it fun?

Tell students that video games are made by teams of people with different roles and skills, working together. For example, game designers figure out how the game will be played, visual designers create the look and feel of the game, experience designers make sure it’s easy to understand, and software engineers, or programmers, actually build the game in the computer. Today students will play the roles of programmer and game designer as they build their own arcade game in CodeCombat. They'll be looking at several types of arcade games, then building their own from scratch.

Explain

For students with no prior coding experience, you should describe some basic concepts before they begin. Project the first level of the activity and use it as an example as you explain:

- **Programs** are sets of instructions for a computer. The computer follows them in the exact order they're written and doesn't do anything that is not in the program, so every single step must be carefully described.

- Programs are written in **programming languages**. Each language has its own **syntax**, which is the rules about capital letters, indentation, punctuation, and so forth. These rules must be followed exactly or the computer won't be able to read the code.
- Many commands include an action, called a method, and some additional instructions, called arguments. Arguments tell the details, like what to perform the action on or where to perform it. For example, in `game.spawnPlayerXY("raider", 40, 34)` , we have `spawnPlayerXY` as the method, and `"raider"` , `40` , and `34` as arguments telling the type of player to spawn and the coordinates where it should appear.
- **Objects** in the program have **properties**, such as speed and maxHealth. Some methods allow the student to change these properties. Samples are shown in the starter code.
- **Comments** are lines in the code that the computer doesn't read. They explain what the code does. Comment lines begin with # in Python, and // in JavaScript.

You may also want to point out the following features on the screen:

- The **blue pop-up box** contains important instructions and explanations.
- The **starter code** also contains important information.
- The **Hints** button in the top right can be helpful if they get stuck.
- the **Methods** section in the middle of the screen provides a bank of commands. Click a command to see more information.

Coding Time (40 mins)

Hand out copies of the Syntax Guide for students to use as a reference.

Have students open the activity and allow them to move through the levels at their own pace. Circulate and assist as they work, calling attention to the Hints button in the top right corner of each level as needed.

When students reach the last level, suggest that they copy and paste code snippets from prior levels, remixing and modifying as needed to build their own game.

Closure (5 mins)

Have students trade computers and play one another's games. Then choose from these activities to facilitate reflection:

- Have students give each other 2 pieces of positive feedback and 1 piece of constructive feedback on their game.
- Facilitate a class discussion using one or more of these questions:
    - What was the most exciting or interesting thing you did today? What was the most challenging?
    - What do you think it would be like to be a game designer or a programmer? Which would you rather be, and why?