Making Game Images

(Also available in WeScheme)

Students choose, import, scale and orient images for their game, practicing reading comments to make sense of and begin to edit a large body of code.

Lesson Goals	Students will be able to: - Apply previous knowledge of ***functions*** to new situations - Use reasoning skills to select appropriate functions and combine their effects
Student-Facing Lesson Goals	- Let's use ***functions*** to transform ***images*** so that we can use them in our games!
Prerequisites	- Simple Data Types - Contracts - Defining Values
Materials	- PDF of all Handouts and Page - Blank Game Starter File - Lesson Slides - Printable Lesson Plan (a PDF of this web page)
Supplemental Materials	- Additional Printable Pages for Scaffolding and Practice
Preparation	Students should have their completed "Game Design" work.

Key Points for the Facilitator	• Discuss copyright and fair use guidelines with your students. • Instructional time may vary based on students' experience with using Google Image Search. • Check beforehand for any issues the school Internet security blocker might cause with searching for images. • There are two ways of importing images: linking directly to the image on the web or downloading the image to Google Drive and then using the "Insert" button. An advantage to working with images from Google Drive is that your game will work in perpetuity, regardless of what happens to the image on the web. See the "Linking Images Guide" below for more information on linking directly. • Encourage students to focus on finding and scaling each image as needed before moving on to the next one.

Glossary

define :: to associate a descriptive name with a value

function :: a relation from a set of inputs to a set of possible outputs, where each input is related to exactly one output

image :: a type of data for pictures

scale :: resize an image to be larger or smaller while maintaining ratios and proportions

string :: a data type for any sequence of characters between quotation marks (examples: "hello", "42", "this is a string!")

The Game Starter File

15 minutes

Overview

This activity is primarily about *review and reading comprehension*, in which students open a large and unfamiliar file and must make sense of it using what they've seen before.

Launch

By now you've learned about defining values, composing functions, and reading contracts. Taken together, that's a lot of code you're now able to understand! It's time to flex your reading skills, and look at the file you'll be working with to build your video game.

This file has code you haven't seen before! And that's ok! For now, see what parts you recognize, and make sure you understand them.

Investigate

- With your partner, load the Blank Game Starter File.
- As you investigate the starter file, record what you Notice and Wonder on Notice and Wonder.

Synthesize

- What familiar things did you see in the Game Starter File file?
- What were some unfamiliar things? Any idea what they might do?
  - *Answers vary: new functions, comments, images*
- What data type is `GAME-TITLE`? What data type is `BACKGROUND`?
  - `GAME-TITLE` *is a String,* `BACKGROUND` *is an Image*
- What does `SCREENSHOT` return in the **Interactions Area**?
  - *An image of the* `BACKGROUND`, `PLAYER`, `TARGET`, *and* `DANGER` *all together*
- Did anyone try pressing "Run"? What happens when you press "Run"?
  - *Allow students to discuss what they see and what connections they see with the code*
- What do you think `image-url` does?
  - *Answers vary: It consumes a **String**, which is a URL (an image location on the Internet) and produces the **Image** inside our program*

What is SCREENSHOT?

The Game Starter File defines several image values, such as `BACKGROUND`, `PLAYER`, etc. These definitions are using the running game, which appears when you click "Run". `SCREENSHOT` is defined as a fixed composition of the game images, placing each of them on top of the background at various (x,y) coordinates. It is used to give students a chance to see their characters onscreen before they've gotten them moving, and to give teachers an opportunity to review coordinates. Other than the ***scaling***, it is *not* in any way connected to the running game, so changes made to `SCREENSHOT` will not impact the game that appears when clicking "Run" and the location of the characters on the background in the screenshot is random and irrelevant.

Finding Your Game Images *flexible*

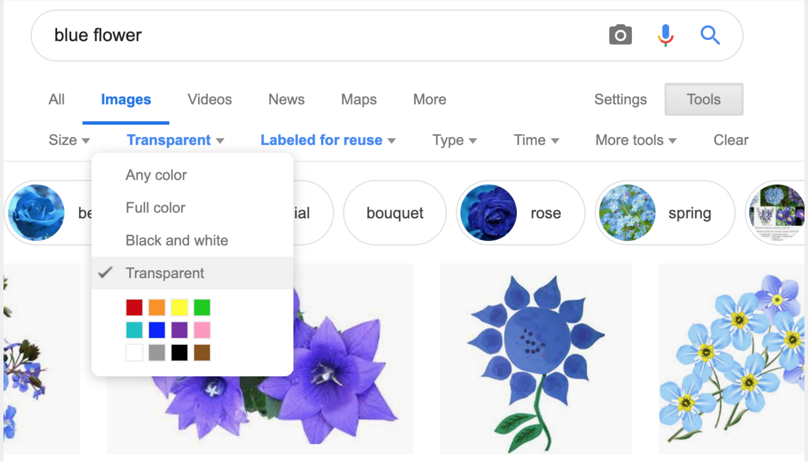
Overview

This activity is all about finding the right images for students' games. Since the internet never has *exactly* the right image, students' need to get their games **just right** motivates them to confront the need for dilation, rotation, and reflection of the images they find. This, in turn feeds back into their understanding of Contracts and Function Composition.

Launch

Guide the students through finding an image, saving it to their Drive, importing it into their program, and defining the image value as `PLAYER`. *Students will change this image later on their own, this is just for teaching purposes.*

- In your favorite search engine (we recommend DuckDuckGo), search for an image and then click "Images".
- Click "All Types" and select "Transparent". If you're using Google Image Search, select "Color -> Transparent". This will filter and display images that have a transparent background, appearing as a light white/grey checkerboard pattern behind the character.



- Once an image has been selected, click it to expand and save the image to Google Drive. For file management, students may want to create a folder to store their game images.
  - If using a Chromebook, this is done by right-clicking and selecting "Google Drive" on the left for the save location.
  - On a PC or Mac, follow this Quick Guide to Saving Images to Drive.

Once the image is saved to Google Drive, it can be brought into the program by using the "Insert" button. This will automatically bring in the image using the `image-url` function, and students can run the code to see the image.

Investigate

What happens if the image we find needs to be made bigger or smaller?

- Students can use the `scale` function to resize their image.
- If your class could use some practice with scaling first, turn to Scaling Practice and/or Scaling Practice (2).

What if the image needs to be rotated, or flipped?

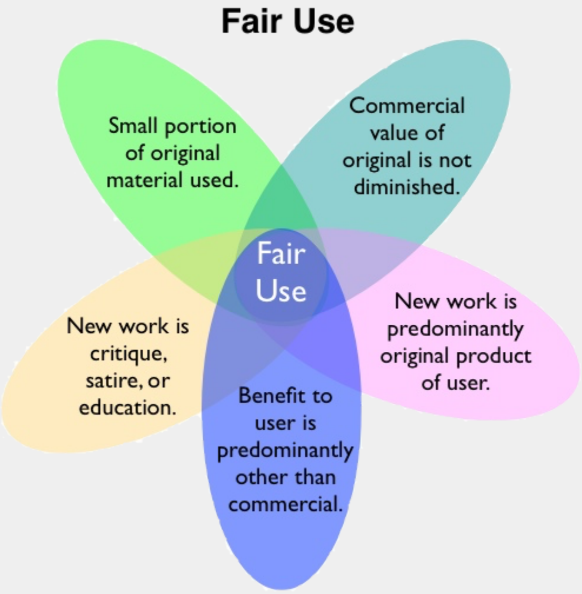
- Students can make-use of the image manipulation functions `rotate`, `flip-horizontal`, and `flip-vertical`. You'll find their contracts in the Contracts Page.

Strategies for English Language Learners

MLR 8 - Discussion Supports: As students discuss, rephrase responses as questions and encourage precision in the words being used to reinforce the meanings behind some of the functions, such as `scale` and `flip-horizontal`.

With your partner, search the Internet for images to use in your game. You will need 4 images, one for each visual element of their game: `BACKGROUND`, `PLAYER`, `DANGER`, `TARGET`

Copyright and Fair Use


Fair Use
Small portion of original material used.
Commercial value of original is not diminished.
Fair Use
New work is critique, satire, or education.
New work is predominantly original product of user.
Benefit to user is predominantly other than commercial.

The students will be using images from the Internet for their game, and while this falls entirely under the "Educational Use" umbrella of Fair Use Guidelines, it is still important to make sure students of all ages understand the purpose of copyright law and the differences between educational and commercial purposes.

When adding an image to their game, students *must* include a comment which gives attribution to the source of the image.

Students should:

- Save the chosen images to their Drive
- Bring them into the programming environment
- Include a comment which gives attribution to the source of the image
- ***Define*** the images as values
- Plan out how to resize and reorient them in their game
- Make sure the final version of each image is defined as either `BACKGROUND`, `TARGET`, `DANGER`, or `PLAYER`

When finished, students should be able to type `SCREENSHOT` in the interactions window and see all four of their images appropriately sized and oriented.

Synthesize

- What functions were most useful in helping you customize your images to make your game look and feel how you want it?
- How did you make use of function composition in customizing your images?