

Making Game Images

(Also available for [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	<p>Students will be able to:</p> <ul style="list-style-type: none"> • Apply previous knowledge of <i>functions</i> to new situations • Use reasoning skills to select appropriate functions and combine their effects
Student-Facing Lesson Goals	<ul style="list-style-type: none"> • I can use different functions to transform <i>images</i>. • I can write definitions for my transformed images.
Materials	<ul style="list-style-type: none"> • Lesson slides • Linking Images Guide - PDF • Blank Game Starter File (Pyret) • <i>Optional:</i> quick guide to saving images to drive (Page #t)
Preparation	<ul style="list-style-type: none"> • Make sure all materials have been gathered • Decide how students will be grouped in pairs
Key Points for the Facilitator	<ul style="list-style-type: none"> • 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. 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.

Language Table

Types	Functions	Values
Number	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>num-expt</code> , <code>num-sqr</code> , <code>num-sqrt</code>	<code>4</code> , <code>-1.2</code> , <code>2/3</code> , <code>pi</code>
String	<code>string-length</code> , <code>string-repeat</code> , <code>string-contains</code>	<code>"hello"</code> , <code>"91"</code>
Boolean	<code><</code> , <code><=</code> , <code>></code> , <code>>=</code> , <code><></code> , <code>==</code> , <code><></code> , <code>>=</code>	<code>true</code> , <code>false</code>
Image	<code>star</code> , <code>triangle</code> , <code>circle</code> , <code>square</code> , <code>rectangle</code> , <code>rhombus</code> , <code>ellipse</code> , <code>regular-polygon</code> , <code>radial-star</code> , <code>text</code> , <code>overlay</code> , <code>above</code> , <code>beside</code> , <code>rotate</code> , <code>scale</code> , <code>flip-horizontal</code> , <code>flip-vertical</code>	

Click here to see the [prior unit-based version](#).

Glossary

define :: to associate a descriptive name with a value

function :: a mathematical object that consumes inputs and produces an output

image :: a type of data for pictures

interactions area :: the right-most text box in the Editor, where expressions are entered to evaluate

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

Students should have their workbook, pencil, and be logged into code.pyret.org and have their completed "Game Design" worksheet.

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 their partner, students should load the [Blank Game Starter File \(Pyret\)](#).

Notice and Wonder

As students investigate the Game Starter File file with their partner, ask students to record what they Notice, and then what they 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

Copyright and Fair Use

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.

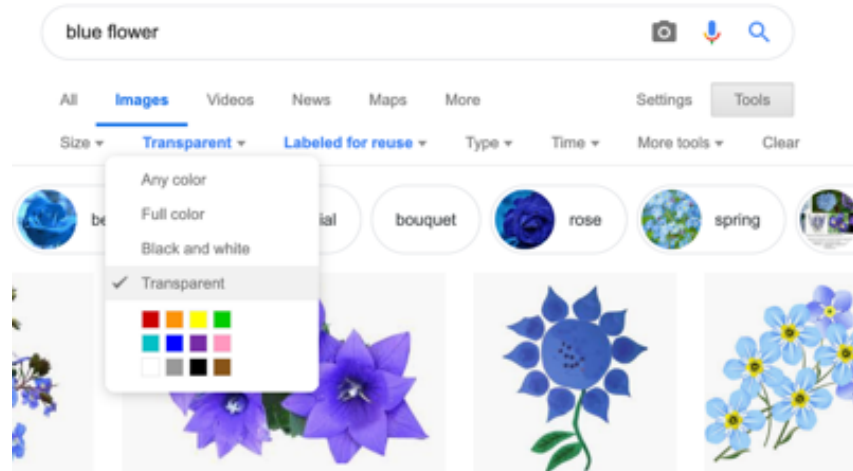
Fair Use

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

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.*

How to find and save images to Google Drive....

In your favorite search engine (we recommend [DuckDuckGo](#)), search for an image and then click "Images". Click "All Types" and select "Transparent" (In Google Image Search, it's under "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 \(Page #t\)](#).

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? What if it needs to be rotated, or flipped?

Students can define the image as a value and make changes to it with the image manipulation functions `scale`, `rotate`, `flip-horizontal`, and `flip-vertical`.

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 their partner, students search the Internet for images to use in their game. They will need 4 images, one for each visual element of their game:

- BACKGROUND
- PLAYER
- DANGER
- TARGET

Students should:

- Save the chosen images to their Drive
- Bring them into the programming environment
- **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.

These materials were developed partly through support of the National Science Foundation, (awards 1042210, 1535276, 1648684, and 1738598). Bootstrap:Algebra by the [Bootstrap Community](#) is licensed under a [Creative Commons 4.0 Unported License](#). This license does not grant permission to run training or professional development. Offering training or professional development with materials substantially derived from Bootstrap must be approved in writing by a Bootstrap Director. Permissions beyond the scope of this license, such as to run training, may be available by contacting contact@BootstrapWorld.org.

