

Assignment 5: Adding Assets

Due 4/18, 2016 10:00am

1 Overview

In this assignment you will be replacing the horribly boring colored squares with image assets. This game is quickly starting to shape up!

2 What you need to do

(1) Download the assets on D2L and import them into your Eclipse project. The assets be placed as a *sibling* of the `src` directory. Remember to *refresh* the project directory in the package explorer after you change files in the project outside of Eclipse.



(2) Load the images into the relevant object files (e.g. load the ghost images in the `Ghost` class). The static `ImageIO.read` method will create an `Image` instance from a `File` instance or throw an `IOException` if the file is not readable (or not an image). You will need to deal with the possibility of an exception. You should **not** load any image file more than once (e.g. loading the file each time you draw a ghost, loading the file in the constructor of an entity, etc). *You will find static blocks useful for this purpose.*

(3) Replace the code that draws colored rectangles with code that draws the newly loaded assets. You can place an image onto a graphics object by calling the `drawImage` method. This method takes the parameters (*image*, *x*, *y*, `null`), where (*x*, *y*) is the upper left corner of the image. Images can be altered with *affine transformations* such as scaling, rotating, and translating. However, the assets are already the correct size for this game and no transformation is necessary.

(4) The player's `draw` method will contain some additional logic. Each invocation of this method will choose one of the eight assets associated with Pacman to draw based on (a) the player's *facing* and (b) an animation counter. Every *ten* draw calls should toggle the player's asset between open-mouth and closed-mouth (*munch* *munch!*). You will need additional instance variables (one which you increment each draw call). You should **not** animate the player if they are not moving (you should store the previous pixel coordinates to test this).

3 Submission

Create a *zip archive* of your Eclipse project (including all template files) and upload it to the correct D2L dropbox before the due date. Again, no late work will be accepted.