Assignment 6: Loading Levels

Due 4/25, 2016 10:00am

1 Overview

In this assignment you will be adding the ability to load level specifications from a file. This assignment will focus on reading content from a file and detecting errors in self-specified ASCII formats (validation errors). You will be throwing and catching specific errors from within the level loading routine.

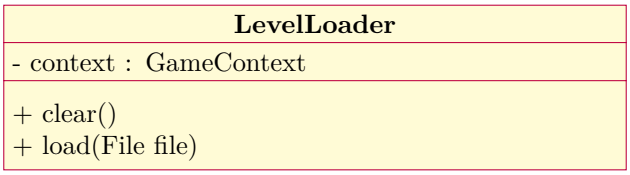
2 What has changed

The GUI driver now has a new menu *Load* which asks for an input file. This action will (after your changes) clear the game context and load the properties of the level as described in the file. The `newGame` method in the driver now takes a file which describes the level (instead of building it statically).

3 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) Implement a `LevelLoader` class according to the following UML diagram.



The `clear` method should remove all entities from the context's entity bag. The `load` method should read the given file, the contents of which is assumed to represent a map, and set properites of the context appropriately. Each character in the map represents a different object:

1. '`#`' represents a wall
2. '`.`' represents a small pellet
3. '`o`' represents a large pellet
4. '`*`' represents a player
5. '`^`' represents a ghost

The initial location of an entity is the row and column where the character is located (e.g. the ASCII version of the map is a direct grid representation of the level). Whitespace should be ignored. In addition to creating entities and adding them to the entity bag, you will also need to set the *size* of the grid (the max x and max y coordinate used), and the player.

(3) There are several exceptional conditions you should check. Each of these conditions should result in a `RuntimeException` with a specific error message.

1. Input file does not exist or is not readable
2. An unknown character is encountered while parsing
3. There are fewer/more than one player on the grid
4. There are fewer/more than two ghosts on the grid

The last two conditions should have two distinct error messages (one for fewer and one for more). You should find the location where a level is loaded and insert *try* blocks around the appropriate areas. When one of the exceptions above occurs, you should display a message to the user. You can do this by using the following (assuming `this` refers to the instance of `Pacman`).

```
JOptionPane.showMessageDialog(this, "Message you'd like to display.");
```

4 File Example

```
####################
#o...#........#...o#
#.##.#.######.# ##.#
#.#   .      .   #.#
#.# ## ##  ## ## #.#
#.     # ^^ #     .#
#.# ## ###### ## #.#
#.#   .  *   .   #.#
#.## #.######.# ##.#
#o...#........#...o#
####################
```

5 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.