Assignment 8: Delays and Pause

Due 5/9, 2016 10:00am

1 Overview

In this assignment you will be adding player invincibility. You will also be adding a few *delays* into gameplay (e.g. a two-second pause when you eat a large pellet).

2 What you need to do

(1) You will need to add the following methods to `GameContext`:

```
public void pauseFor(int delay);
public boolean isInvincible();
public boolean isFlashing();
public void setInvincible(int delay);
```

The first method will simply pause the context (set a variable denoting a paused state), then start a *delay*-second (non-repeating) timer which will un-pause the context when the timer fires. This method has an *asynchronous side-effect*, and when called will *immediatley* pause the game, and then un-pause the game in a background thread.

The `isInvincible` and `isFlashing` methods are simple accessors to new instance variables (which you must create). A context is in *invincible* mode if the player is not in danger from ghosts. A context is in *flashing* mode if the *invincible* state is about to expire. Let's animate this!

The `setInvincible` method should initially set the context to *invincible* but **not** *flashing*. After three-quarters of the *delay* value passes, the context should additionally be set to *flashing*. Then, after the remaining one-quarter of the *delay* value passes, the context should be set to **not** *invincible* and **not** *flashing*.

This method is best done with two timers (where the second time starts *after* the first timer fires). You will also need to ensure that if you call `setInvincible` while another call is still active, it should simply *reset* the call (e.g. eating another large pellet shoudl reset the time from the beginning). An easy way to do this is to *cancel* any active timers from a previous call (*how do you access these objects?*).

(2) After a player-large pellet collision, the context should be set to invincible for *ten* seconds.

(3) There are two assets (scatter and flashing) which are unused that both apply to ghosts. These will need to be loaded in the same manner as the other assets. If the context is *invincible*, then you should use the *scatter* asset for the ghost. If the context is also *flashing*, then you should flip between the *scatter* and *flashing* assets sever *ten* updates using the same animation technique that is used in the player render method. If the context is not *invincible*, then the original drawing behavior should apply.

(4) After a player-ghost collision, the game should pause for one second. If the game context is in *invincible* mode, then the ghost should be removed from the game and 200 points should be added. Otherwise, the original behavior should apply. The ghost *behavior* must also change in a bit of a subtle way. When choosing a direction to move, the ghost should target the square $(0, 0)$ instead of the player's current position if the context is *invincible*. The original behavior should apply when the context is not *invincible*.

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.