SWE-265P: Reverse Engineering and Modeling

Armin Hamidi

1/21/2020

JPACMAN3 Questions

1. What is the role of EmptySprite?

   EmptySprit is a placeholder class with empty methods mimicking the functionality of the working sprites (like AnimatedSprite and PacmanSprite). Below we see EmptySprite's empty methods

```
@Override
public void draw(Graphics graphics, int x, int y, int width, int height) {
    // nothing to draw.
}

@Override
public Sprite split(int x, int y, int width, int height) {
    return new EmptySprite();
}

@Override
public int getWidth() {
    return 0;
}

@Override
public int getHeight() {
    return 0;
}
```

Compared to AnimatedSprite's similar, and functional methods below

```
public void draw(Graphics graphics, int x, int y, int width, int height) {
    update();
    currentSprite().draw(graphics, x, y, width, height);
}

@Override
public Sprite split(int x, int y, int width, int height) {
    update();
    return currentSprite().split(x, y, width, height);
}
@Override
public int getWidth() {
    assert currentSprite() != null;
    return currentSprite().getWidth();
}

@Override
public int getHeight() {
    assert currentSprite() != null;
```

2. What is the role of MOVE_INTERVAL and INTERVAL_VARIATION?

```java
/**
 * The variation in intervals, this makes the ghosts look more dynamic and
 * less predictable.
 */
private static final int INTERVAL_VARIATION = 50;

/**
 * The base movement interval.
 */
private static final int MOVE_INTERVAL = 250;

/**
```

MOVE_INTERVAL sets the base movement interval for each ghost, in milliseconds used by the method seen below.

```java
/**
 * The time that should be taken between moves.
 *
 * @return The suggested delay between moves in milliseconds.
 */
public long getInterval() {
    return this.moveInterval + new Random().nextInt(this.intervalVariation);
}
```

INTERVAL_VARIATION creates a random variation for the possible move interval that the ghost will experience. At this time all their MOVE_INTERVALs are set to 250 with a variation of 50. Meaning each ghost will move every 200-300ms.

3. If you wanted to add fruit, what files would you change?

ImageSprite would be a good class to modify to represent the fruit sprite. PlayerCollisions, Level, and LevelFactory would all have to be modified to account for a new NPC, in this case static fruit. However the fruit would be nearly identical to a pellet, we would have to change their image and rate of appearance.

```java
/**
 * The NPCs of this level and, if they are running, their schedules.
 */
private final Map<Ghost, ScheduledExecutorService> npcs;

/**
```

“their schedules” would be the intervals at which the fruit would appear and disappear.

Below we see the Pellet class which contains almost all the same functionality as a fruit. We can change the image and value right here in this class. However we would need to implement a new method to account for its visibility.

```java
/**
 * A pellet, one of the little dots Pac-Man has to collect.
 *
 * @author Jeroen Roosen
 */
public class Pellet extends Unit {

    /**
     * The sprite of this unit.
     */
    private final Sprite image;

    /**
     * The point value of this pellet.
     */
    private final int value;

    /**
     * Creates a new pellet.
     * @param points The point value of this pellet.
     * @param sprite The sprite of this pellet.
     */
    public Pellet(int points, Sprite sprite) {
        this.image = sprite;
        this.value = points;
    }

    /**
     * Returns the point value of this pellet.
     * @return The point value of this pellet.
     */
    public int getValue() {
        return value;
    }
    @Override
    public Sprite getSprite() {
        return image;
    }
}
```