Useless Fact of the Day

- There are actually about 1,300 varieties of food products made by Heinz and Heinz subsidiaries, including 108 varieties of baby food and 60 kinds of pickles.
- Even when Henry J. Heinz thought up the “57 varieties” slogan, Heinz was making over 60 varieties of things -- “57” was just a random number that Heinz though sounded good.

Using FANG’s Alarm

- Four steps involved:
  - 1. Define a tiny class which implements Alarm
  - note: you can actually put this class *inside* the body of the class you want to schedule the timer from (a GameLevel or GameLoop class), which will let the timer access all the game/level’s Sprites, etc.

```java
public class Timer implements Alarm {
    public void alarm() {
        System.out.println("the alarm went off!");
    }
}
```
Using FANG’s Alarm

• 3. In the class where you want the alarm (the class holding the main game or a level, probably), declare and initialize an instance of this class

```java
/** ... */
/** number of asteroids that have been shot */
private int numAsteroidsShot;
/** the alarm timer */
private Timer timer;
/** ... */
public void startGame ()
{
    timer = new Timer();
    // ...
}
```

Using FANG’s Alarm

• 4. Call `scheduleRelative(theTimer, theTime)` or `scheduleAbsolute(theTimer, theTime)`

```java
// this will cause the timer to go off 3.3 seconds after this code runs
scheduleRelative(timer, 3.3);

// this will cause the timer to go off 3.3 seconds after the beginning of the game (so long as this code runs before that point)
scheduleAbsolute(timer, 3.3);
```

Making a Timed Splash Screen

• 1. Declare and initialize relevant sprites (for a message, or a picture, or a transparent square color, etc.)

• 2. Set the splash screen sprites to visible when you want to show the splash screen

• 3. Schedule a relative timer at the same place as #2 for however many seconds you want the splash screen to remain on-screen

• 4. In the `alarm()` method in your timer, set the splash screen stuff to invisible (we get rid of the splash screen when the alarm goes off)

Making a Countdown Timer

• You can schedule a timer from within its own `alarm()` method! See example on next page!

• By rescheduling the alarm when the alarm goes off, you can have the alarm go off every 2 seconds, or at whatever interval you desire

• To keep from rescheduling forever (or until the end of the game, anyway), you can just add a condition controlling the reschedule (only reschedule if the player’s time left is still greater than zero, or something)
Making a Countdown Timer

- Example:

```java
public class Timer implements Alarm {
    public void alarm() {
        // reschedule myself for 1 second later
        scheduleRelative(this, 1.0);
        // print something for testing
        System.out.println("this will print out once every second");
    }
}
```

- **Note:** You also have to *initially* schedule the timer from somewhere outside of its own `alarm()` method, to start the self-rescheduling process.