Lab 3 posted this week

Friday, March 26\textsuperscript{th} – Review for Exam 3

Monday, March 29\textsuperscript{th} – Exam 3

Wednesday, March 31\textsuperscript{st} – Go over Exam 3

Friday, April 2\textsuperscript{nd} – Class cancelled

(Adrienne will be out of town April 1\textsuperscript{st} – 4\textsuperscript{th})
CHAPTER 7 - DRAWING STARS

- Drawing stars on the screen
  - Create method for drawing stars and call it from constructor of Space
  - Inside method we retrieve the background image and draw ovals at random locations
  - We also added functionality to create stars in random shades of gray.

SPEEDING UP ROCKET

- Create code so that the rocket will show a different image when the user selects to speed it up.
Recall from earlier examples the following code:

```java
Actor a = getOneIntersectingObject(X.class);
```

Remember that X is the class we are interested in looking for collisions with – it can be anything (Flower, Ball, Brick, Barrel).

`getOneIntersectingObject` returns the object we are interesting with or null if not intersecting an object of the passed-in type. The object that is passed back is of type Actor.

Therefore, the type of the variable `a` is `Actor`.

If we try to do this:

```java
X a = getOneIntersectingObject(X.class);
```

The code will not compile because `getOneIntersectingObject` returns an Actor, not an X.

But we know that the Actor that is really being returned is an X.
However, sometimes we may want to do things with a (the variable) that only X's can do.

However, a is an Actor and can only do things Actors can do.

If we want to treat the object that is returned by `getOneIntersectingObject` as an X, we can explicitly cast it as an X.

```java
X a = (X) getOneIntersectingObject(X.class);
```

The `(X)` is the cast.
In ProtonWave class, we see a number of new things:
- Array
- While loop

Each of these things is explained in greater detail in Chapter 5. We are not covering the example from Chapter 5, but these concepts are being covered.

Arrays

- Another type of collection (way to keep track of a group of objects).
- Arrays are fixed size.
- To declare a variable that holds an array:
  
  ```java
  TypeOfThingInArray[] name;
  ```

- To create an array and assign it to the variable:
  
  ```java
  name = new TypeOfThingInArray[NUMBE]R;
  ```
  
  Where number is the number of elements you can store in the array.
**ARRAYS**

- You can access elements in an array by using their index.
- Indices for an array are from 0 to size -1. So, if there are 20 elements in an array, valid indices are 0-19.
  
  ```
  ArrayName[ index ]
  ```

  Would allow you to access the element at that index

  ```
  ArrayName [ index ] = blah;
  ```

  Would assign `blah` to that index.

**WHILE-LOOP**

- Another form of iteration (looping).
- This loop is not a counting loop like the for-loop, but rather will keep looping until the condition indicated on the loop is false.

  ```
  while ( booleanExpression )
  {
    // code that should be repeated
  }
  ```