Assignment 05

- need headers in all files
- Make sure you do not rely on defaults
- Restore defaults if you change them!
  - `rectMode`
  - `ellipseMode`
  - `strokeWeight`
  - transformations
- Using transformations to draw means x and y should not appear as coordinates
- Do NOT override/mask fields inherited from superclass
- parameters passed in through the constructor(s) are supposed to affect your drawing
- When 50 is given, a creature of 50x50 pixels is supposed to be drawn

Assignment 05

- in react(), check for
  - `distance < size+objs[i].getSize()`
  - not `< size*2`
- If you really want to draw bigger/smaller, at the minimum you need to overwrite `getSize()` so that it returns the appropriate size for your creature

Assignment 06

- Do not write the processed images as 4 files then load them.
- Create images in memory and manipulate them there instead.
  - `loadImage()` into 4 different `PImage` objects
  - `createImage()`
- Remember that your collage can contain multiple copies of the same image(s), it doesn't just have to be the 4.
- The requirement of sketch window size is relaxed
  - however, you should have a default size in mind, and scale appropriately if it is changed.

Creating strings

- Strings - `"a", "abc"`
- Characters - 'a'
- Declaring String objects

```
String myName;
```

- Declaring String objects with initialization

```
String myName = "Fred";
String myName = new String("Fred");
String myName = new String(); // empty
  String
```

String class methods

- `charAt(int` *`index`*`)`
  - Returns the character at the specified index
- `equals(String` *`anotherString`*`)`
  - Compares a string to a specified object
- `equalsIgnoreCase(String` *`anotherString`*`)`
  - S/A ignoring case (i.e. 'A' == 'a')
- `indexOf(char` *`c`*`)`
  - Returns the index value of the first occurrence of a character within the input string
- `indexOf(String` *`str`*`)`
  - Returns the index value of the first occurrence of a substring within the input string
- `length()`
  - Returns the number of characters in the input string
- `substring(int` *`startIndex, int endIndex`*`)`
  - Returns a new string that is part of the input string
- `toLowerCase()`
  - Converts all the characters to lower case
- `toUpperCase()`
  - Converts all the characters to upper case
- `String concat(String` *`anotherString`*`)`
  - Concatenates with anotherString and returns it

String functions

```
String str = "Roses are red";

println(str.length());
println(str.toUpperCase());

str = str.concat(", Violets are blue");
//str = str + (", Violets are blue");
println(str);

char c = str.charAt(3);
println(c);

for (int i=0; i<str.length(); i++) {
  println(str.charAt(i));
}

println(str.indexOf('e'));
```

String equality

- use `.equals()` to compare the contents of two strings
- not ==
  - Strings are objects
  - == compares whether two string objects are the same

```
String str = new String("one");
String str2 = new String("one");

println(str == str2);
println(str.equals(str2));
```

Implement equals() and indexOf()

- Write a function that takes two strings and returns `true` if the two strings have the same content and `false` otherwise.
- Similarly, implement indexof
- How would you implement a function that counts the number of times the vowel 'e' (or any other char) appears in a string?

Text Display

- textLetter
  - Prints out text character by character, changing to a random font size every time.
- textHeadline
  - scroll text from right of screen to left of screen and wrap text
- textRotate
  - rotate text in 3D
- textCrawl
  - scroll text like star wars