Breakout Lesson 5

Adding Levels

More Levels

You need to create at least two more methods in your world that lay bricks out in different patterns.

We will modify the act method from the Adding Title slides so that:

- The next level will load if the current level has been cleared.
- Your game should have at least 3 unique levels
- Set the ball centered on top of the paddle at the start of each level
- When all levels have been cleared, "You Win" will be displayed (state will be changed to WIN)
- If the state is WIN and space is pressed, remove "You Win", add the title and set state to MENU

```
public void act() {
      if (state == MENU) {
           if (user presses space key) {
                // Change state to PLAYING and set up the World by
                // removing the Title and adding a Ball and Paddle
           }
      }
      else if (state == PLAYING) {
           if (there are no bricks present) {
                increment level counter
                Set the Ball on top of the Paddle (it starts at the paddle each level)

                if (level == 1) {
                     // Make level 1
                } else if (level == 2) {
                     // Make level 2
                } else if (level == 3) {
                     // Make level 3
                } else {
                      // Change state to WIN
                     // Add a "You Win" object to the screen
                     // Remove paddle, ball
                }
           }
      } else if (state == WIN) {
           if (user presses space key) {
              // remove "You Win", add the title and change state to MENU and level to 0
           }
```

How to tell when there are no more Bricks

```
java.util.List getObjects(java.lang.Class cls)
               Get a List of all objects in the World of a particular class.
```

This function returns a List of all objects in the World of a particular class. In order to use List objects, you must import the List class. Add the following statement below "`import greenfoot.*;`" at the very top of your World code.

```
import java.util.List;    // Add to top of your World code
```

A `List` is just like an array in Javascript. Here are some `List` methods from the `List` API:

```
void add(int index, E element)
      Inserts an element at the desired index position
E get(int index)
      Returns the element at the specified position
int size()
      Returns size of the list
```

To tell when there are no more Bricks in the World, use the World method `getObjects()` and save the return value into a List variable. Then ask the List for its size. If the size is zero, there are no more Bricks.

If your ball doesn't break Bricks

For now, instead of having the ball bounce off of bricks before destroying them, just destroy any Bricks the ball touches.

Add to your Ball code:

```
Actor brick = getOneIntersectingObject(Brick.class)
if (brick != null) {
    getWorld().removeObject(brick);
}
```

Extensions

- Customize the Title / Menu screen.
- Customize the "You Win" screen.
- Add a score.
- Add feature so "Press SPACE to start" is displayed after a level is drawn. The level doesn't begin until the user presses SPACE.
- Add a "Pause" feature such that pressing "p" pauses/unpauses the game