Assignment 6: “Angry Birds, The Tri-Shot Update”

Objectives

- Read and understand all of the Angry Birds source code from Lecture 6.
- Implement it such that when the player presses the space bar after they’ve launched an `Alien` (and it hasn’t hit anything yet), split the `Alien` into three `Aliens` that all behave just like the base `Alien`.

Getting Started

Download the distro code for your game from https://cdn.cs50.net/games/2019/x/assignments/6/assignment6.zip and unzip `assignment6.zip`, which should yield a directory called `assignment6`.

Then, in a terminal window (located in `/Applications/Utilities` on Mac or by typing `cmd` in the Windows task bar), move to the directory where you extracted `assignment6` (recall that the `cd` command can change your current directory), and run

```
cd assignment6
```

Three’s Company

Welcome to your sixth assignment! This week, we took a look at the fundamentals of Box2D, one of the most widely-used 2D physics engines, and how it ties into LÖVE, with its built-in wrappers for it. This assignment will be a little simpler than some of the previous ones (indeed, there’s only one core objective, albeit a reasonably complex one) but will still require knowledge of Box2D and the distro before we can dive in too quickly.

Your goal this assignment:

- *Implement it such that when the player presses the space bar after they’ve launched an `Alien` (and it hasn’t hit anything yet), split the `Alien` into three `Aliens` that all behave just like the base `Alien`.* The code for actually launching the `Alien` exists in `AlienLaunchMarker`, and we could naively implement most, if not all, of this code in the same class, since the `Alien` in question we want to split off is a field of this class. However, because we want to only allow splitting before we’ve hit anything, we need a flag that will get triggered whenever this `Alien` collides with anything else, so we’ll likely want the logic for this in the `Level` itself here, since that is where we pass in the collision callbacks via `World:setCallbacks()` . The center `Alien` doesn't really need to be modified for the splitting process; really, all we need to do is spawn two new `Alien` s at the right angle and velocity so that it *appears* we've turned the single `Alien` into three, one above and one below. For this, you'll need to take linear velocity into consideration. Additionally, be aware that the `Alien` we want to launch has the `userData` of the string "Player", as opposed to the `Alien` we want to kill, which has just the `userData` of "Alien". Lastly, be sure that the launch marker doesn't reset until *all* of the `Alien` s we fling have slowed to nearly being still, not just the one `Alien` we normally check. In all, you should have all of the pieces at this point you need in order to make this happen; best of luck!

How to Submit

1. If you haven't done so already, visit this link, log in with your GitHub account, and click **Authorize cs50**. Then, check the box indicating that you'd like to grant course staff access to your submissions, and click **Join course**.

2. Using Git, push your work to `https://github.com/me50/USERNAME.git` , where `USERNAME` is your GitHub username, on a branch called `games50/assignments/2020/x/6` or, if you've installed `submit50` , execute

   ```
   submit50 games50/assignments/2020/x/6
   ```

   instead.

3. Record a 1- to 5-minute screencast in which you demonstrate your app's functionality and/or walk viewers through your code. Upload that video to YouTube (as unlisted or public, but not private) or somewhere else.
4. Submit this form.

You can then go to https://cs50.me/cs50g to view your current progress!