

CS50's

Introduction to Game Development

OpenCourseWare

Colton Ogden (<https://www.linkedin.com/in/colton-ogden-0514029b/>)
cogden@cs50.harvard.edu

David J. Malan (<https://cs.harvard.edu/malan/>)
malan@harvard.edu

f (<https://www.facebook.com/dmalan>) **G** (<https://github.com/dmalan>) **@**
(<https://www.instagram.com/davidjmalan/>) **in** (<https://www.linkedin.com/in/malan/>)
ID (<https://orcid.org/0000-0001-5338-2522>) **Q** ([https://www.quora.com/profile](https://www.quora.com/profile/David-J-Malan)
/David-J-Malan) **5** (<https://www.reddit.com/user/davidjmalan>) **d**
(<https://www.tiktok.com/@davidjmalan>) **📍** (<https://davidjmalan.t.me/>) **🐦**
(<https://twitter.com/davidjmalan>)

Angry Birds

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`.

Demo

by Edward Kang



Getting Started

Download the distro code for your game from cdn.cs50.net/games/2018/x/projects/6/angry.zip (<https://cdn.cs50.net/games/2018/x/projects/6/angry.zip>) and unzip `angry.zip`, which should yield a directory called `angry`.

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 `angry` (recall that the `cd` command can change your current directory), and run

```
cd angry
```

Three's Company

Welcome to your seventh 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.

Specification

- 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

When you submit your project, the contents of your `games50/projects/2018/x/angry` branch must match the file structure of the unzipped distribution code exactly as originally received. That is to say, your files should not be nested inside of any other directories of your own creation or otherwise deviate from the file structure we gave you. Your branch should also not contain any code from any other projects, only this one. Failure to adhere to this file structure will likely result in your submission being rejected.

By way of a simple example, for this project that means that if the grading staff visits `https://github.com/me50/USERNAME/blob/games50/projects/2018/x/angry/src/AlienLaunchMarker.lua` (where `USERNAME` is your own GitHub username as provided in the form, below) we should be brought to the `AlienLaunchMarker.lua` file for Angry 50. If that's not how your code is organized when you check (e.g., you get a 404 error or don't see your edits), reorganize your repository as needed to match this paradigm.

1. If you haven't done so already, visit [this link \(https://submit.cs50.io/invites/46e6f2ea29954ce9bb1bdc478a440055\)](https://submit.cs50.io/invites/46e6f2ea29954ce9bb1bdc478a440055), 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**.

The change to `/projects/2018` below is intentional, as CS50 courses have changed to a scheme that reflects when the project was initially released. So the 2018 here is correct, even though it's no longer 2018! Also note that the **video** for this project is optional, you do not need to complete it if you do not want to. This is only true for this project, Project 0, and Project 7. For all other projects, a video is mandatory.

2. Using [Git \(https://git-scm.com/downloads\)](https://git-scm.com/downloads), push your work to `https://github.com/me50/USERNAME.git`, where `USERNAME` is your GitHub username, on a branch called `games50/projects/2018/x/angry` or, if you've installed `submit50` (<https://cs50.readthedocs.io/submit50/>), execute

```
submit50 games50/projects/2018/x/angry
```

instead.

3. [Record a screencast \(if you want\) \(https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/\)](https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/), not to exceed 5 minutes in length (and not uploaded more than one month prior to your submission of this project) in which you demonstrate your app's functionality. [Upload that video to YouTube \(https://www.youtube.com/upload\)](https://www.youtube.com/upload) (as unlisted or public, but not private) or somewhere else.
4. [Submit this form \(https://forms.cs50.io/ab8879d2-a6e4-42bc-affb-e138a9430ad0\)](https://forms.cs50.io/ab8879d2-a6e4-42bc-affb-e138a9430ad0).

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