

# 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/David-J-Malan>) **5** (<https://www.reddit.com/user/davidjmalan>) **d**  
(<https://www.tiktok.com/@davidjmalan>) **📍** (<https://davidjmalan.t.me/>) **🐦**  
(<https://twitter.com/davidjmalan>)

## Breakout

---

### Objectives

---

- Read and understand all of the Breakout source code from Lecture 2.
- Add a powerup to the game that spawns two extra `Ball`s.
- Grow and shrink the Paddle when the player gains enough points or loses a life.
- Add a locked `Brick` that will only open when the player collects a second new powerup, a key, which should only spawn when such a `Brick` exists and randomly as per the `Ball` powerup.

### Demo

---

*by Edward Kang*



## Getting Started

---

Download the distro code for your game from [cdn.cs50.net/games/2018/x/projects/2/breakout.zip](https://cdn.cs50.net/games/2018/x/projects/2/breakout.zip) (<https://cdn.cs50.net/games/2018/x/projects/2/breakout.zip>) and unzip `breakout.zip`, which should yield a directory called `breakout`.

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

```
cd breakout
```

## Break(out) a Leg!

---

Welcome to your third assignment! By now, we've gotten our feet wet with states, randomization, and much more; this time, we'll be diving in a little bit more and adding some new features entirely!

## Specification

---

- Add a `Powerup` class to the game that spawns a powerup (images located at the bottom of the sprite sheet in the distribution code). This `Powerup` should spawn randomly, be it

on a timer or when the Ball hits a `Block` enough times, and gradually descend toward the player. Once collided with the `Paddle`, two more `Ball`s should spawn and behave identically to the original, including all collision and scoring points for the player. Once the player wins and proceeds to the `VictoryState` for their current level, the `Ball`s should reset so that there is only one active again.

- Grow and shrink the `Paddle` such that it's no longer just one fixed size forever. In particular, the `Paddle` should shrink if the player loses a heart (but no smaller of course than the smallest paddle size) and should grow if the player exceeds a certain amount of score (but no larger than the largest `Paddle`). This may not make the game completely balanced once the `Paddle` is sufficiently large, but it will be a great way to get comfortable interacting with `Quad`s and all of the tables we have allocated for them in `main.lua`!
- Add a locked `Brick` (located in the sprite sheet) to the level spawning, as well as a key powerup (also in the sprite sheet). The locked `Brick` should not be breakable by the ball normally, unless they of course have the key `Powerup`! The key `Powerup` should spawn randomly just like the `Ball` `Powerup` and descend toward the bottom of the screen just the same, where the `Paddle` has the chance to collide with it and pick it up. You'll need to take a closer look at the `LevelMaker` class to see how we could implement the locked `Brick` into the level generation. Not every level needs to have locked `Brick`s; just include them occasionally! Perhaps make them worth a lot more points as well in order to compel their design. Note that this feature will require changes to several parts of the code, including even splitting up the sprite sheet into `Brick`s!

## Errata

- `ParticleSystem:setAreaSpread` has been renamed to `ParticleSystem:setEmissionArea` in LÖVE 11.0.
- `love.filesystem.exists` has been renamed to `love.filesystem.getInfo` in LÖVE 11.0.
- `love.audio.newSource` now must take a second argument (`'static'` or `'stream'`), whereas previously it was optional.

## How to Submit

When you submit your project, the contents of your `games50/projects/2018/x/breakout` 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/breakout/src/Paddle.lua` (where `USERNAME` is your own GitHub username as provided in the form, below) we should be brought to the `Paddle.lua` file for Breakout. 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!

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/breakout` or, if you've installed [submit50 \(https://cs50.readthedocs.io/submit50/\)](https://cs50.readthedocs.io/submit50/), execute

```
submit50 games50/projects/2018/x/breakout
```

instead.

3. [Record a screencast \(https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/\)](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.
  - To aid in the staff's review, in your video's description on YouTube, you should timestamp where each of the following occurs *in your gameplay demonstration*. This is **not optional**; failure to do this will result in your submission being rejected:
  - Multiball powerup
  - Paddle growth
  - Key powerup obtained

- Key block unlocked

4. [Submit this form \(https://forms.cs50.io/0c7f116a-6c1e-433d-a6da-b6d7ee856078\)](https://forms.cs50.io/0c7f116a-6c1e-433d-a6da-b6d7ee856078).

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