Assignment 2: “Breakout, The Powerup Update”

Objectives

- Read and understand all of the Breakout source code from Lecture 1.
- 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.

Getting Started

Download the distro code for your first game from https://cdn.cs50.net/games/2019/x/assignments/2/assignment2.zip and unzip `assignment2.zip` , which should yield a directory called `assignment2` .

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 `assignment2` (recall that the `cd` command can change your current directory), and run

```
cd assignment2
```

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!

Your goals this assignment:

- 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!

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/2019/x/2` or, if you’ve installed `submit50` , execute

   ```
   submit50 games50/assignments/2019/x/2
   ```

   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!