CSCI E-23a

Introduction to Game Development

produced by CS50

Colton Ogden
David J. Malan

Harvard Extension School
Spring 2018

Assignment 3: "Match-3, The Shiny Update"

Objectives

- Read and understand all of the Match-3 source code from Lecture 3.
- Implement time addition on matches, such that scoring a match extends the timer by 1 second per tile in a match.
- Ensure Level 1 starts just with simple flat blocks (the first of each color in the sprite sheet), with later levels generating the blocks with patterns on them (like the triangle, cross, etc.). These should be worth more points, at your discretion.
- Creat random shiny versions of blocks that will destroy an entire row on match, granting points for each block in the row.
- Only allow swapping when it results in a match. If there are no matches available to perform, reset the board.
- (*Optional*) Implement matching using the mouse. (Hint: you'll need `push:toGame(x,y)`; see the `push` library's documentation here for details!

Getting Started

GitHub Classroom

In this course, we'll use GitHub Classroom to distribute projects and collect submissions. To begin Assignment 3:

1. Click here to go to the GitHub Classroom page for starting the assignment.
2. Click the green "Accept this assignment" button. This will create a GitHub repository for your project. Recall that a git repository is just a location where your code will be stored and which can be used to keep track of changes you make to your code over time.
3. Click on the link that follows "Your assignment has been created here", which will direct you to the GitHub repository page for your project. It may take a few seconds for GitHub to finish creating your repository.
4. In the upper-right corner of the repository page, click the "Fork" button, and then (if prompted) click on your username. This will create a fork of your project repository, a version of the repository that belongs to your GitHub account.
5. Now, you should be looking at a GitHub repository titled **username/assignment3-username**, where **username** is your GitHub username. This will be the repository to which you will push all of your code while working on your assignment. When working on the assignment, do not directly push to the **games50/assignment3-username** repository: always push your code to your **username/assignment3-username** repository.

Setup

Time to pull down the starting code for Match-3! First, on your main repository page (https://github.com/username/assignment3-username), click on the green “Clone or download” button. Copy the “Clone with HTTPS” link to your clipboard (if familiar with SSH, you can use that instead).

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 want to store your project on your computer (recall that the `cd` command can change your current directory), and run

```
git clone repository_url assignment3
```

where `repository_url` is the link you just copied from GitHub. You will be prompted for your GitHub username and password

Go ahead and run `cd assignment3` to enter your repository.

A Match (3) Made in Heaven

Welcome to your fourth assignment! There was a lot to learn with timers, tweens, and more in this lecture, but unfortunately, our game is still lacking in a few areas. By extending its functionality, we’ll have something even closer to famous titles such as *Bejeweled* and *Candy Crush Saga*!

Your goals this assignment:

- *Implement time addition on matches, such that scoring a match extends the timer by 1 second per tile in a match*. This one will probably be the easiest! Currently, there's code that calculates the amount of points you'll want to award the player when it calculates any matches in `PlayState:calculateMatches` , so start there!
- *Ensure Level 1 starts just with simple flat blocks (the first of each color in the sprite sheet), with later levels generating the blocks with patterns on them (like the triangle, cross, etc.). These should be worth more points, at your discretion*. This one will be a little trickier than the last step (but only slightly); right now, random colors and varieties are chosen in `Board:initializeTiles` , but perhaps we could pass in the `level` variable from the `PlayState` when a `Board` is created (specifically in `PlayState:enter`), and then let that influence what `variety` is chosen?
- *Create random shiny versions of blocks that will destroy an entire row on match, granting points for each block in the row*. This one will require a little more work! We'll need to modify the `Tile` class most likely to hold some kind of flag to let us know whether it's shiny and then test for its presence in `Board:calculateMatches` !
- *Only allow swapping when it results in a match. If there are no matches available to perform, reset the board*. There are multiple ways to try and tackle this problem; choose whatever way you think is best! The simplest is probably just to try and test for `Board:calculateMatches` after a swap and just revert back if there is no match! The harder part is ensuring that potential matches exist; for this, the simplest way is most likely to pretend swap everything left, right,

up, and down, using essentially the same reverting code as just above! However, be mindful that the current implementation uses all of the blocks in the sprite sheet, which mathematically makes it highly unlikely we'll get a board with any viable matches in the first place; in order to fix this, be sure to instead only choose a subset of tile colors to spawn in the `Board` (8 seems like a good number, though tweak to taste!) before implementing this algorithm!

- *(Optional) Implement matching using the mouse. (Hint: you'll need `push:toGame(x,y)`; see the `push` library's documentation here for details!* This one's fairly self-explanatory; feel free to implement click-based, drag-based, or both for your application! This one's only if you're feeling up for a bonus challenge :) Have fun!

How to Submit

Step 1 of 1

1. Go to the GitHub page for your **username/assignment3-username** repository (note: this is different from the **games50/assignment3-username** repository).
2. On the right side of the screen, click the Pull request button.
3. Make sure that the "base fork" is `games50/assignment3-username`, and the "head fork" is `username/assignment3-username`.
4. Click "Create pull request".
5. On the next page, click the "Create pull request" button again.

Congratulations! You've completed Assignment 3.