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
(https://www.facebook.com/dmalan) (https://github.com/dmalan)
(https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)
(https://orcid.org/0000-0001-5338-2522) (https://www.quora.com/profile/David-J-Malan) (https://www.reddit.com/user/davidjmalan)
(https://www.tiktok.com/@davidjmalan) (https://davidjmalan.t.me/)
(https://twitter.com/davidjmalan)

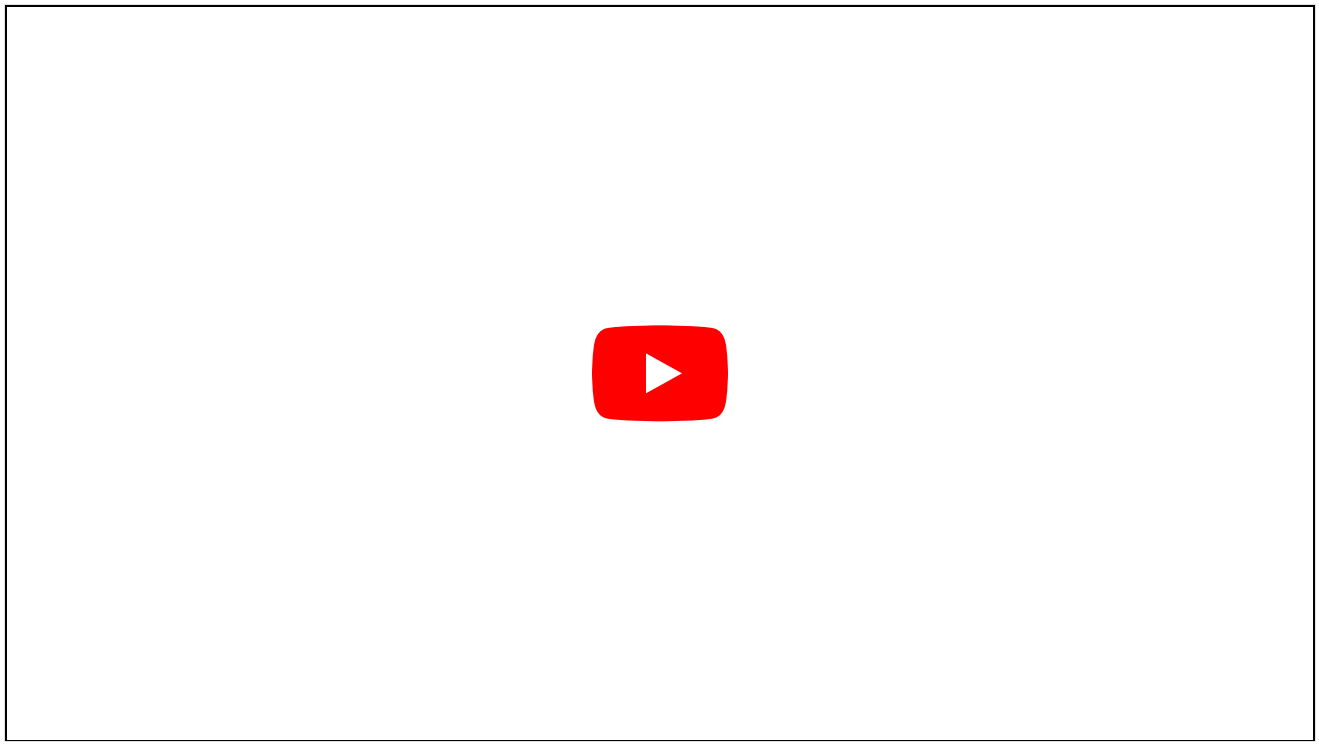
Helicopter

Objectives

- Download Unity and get familiar with its interface.
- Read and understand all of the Helicopter Game 3D source code from Lecture 8.
- Add Gems to the game that spawn in much the same way as Coins, though more rarely so. Gems should be worth 5 coins when collected and despawn when off the left edge of the screen.
- Fix the bug whereby the scroll speed of planes, coins, and buildings doesn't reset when the game is restarted via the space bar.

Demo

by Edward Kang



Getting Started

Download the distro code for your game from cdn.cs50.net/games/2018/x/projects/8/helicopter.zip (https://cdn.cs50.net/games/2018/x/projects/8/helicopter.zip) and unzip `helicopter.zip`, which should yield a directory called `helicopter`.

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

```
cd helicopter
```

Downloading Blender

> Make sure to download the correct (older!) version of Blender, not the latest version, per these instructions!

First, in order to be able to import some models into our scene appropriately (and to hopefully give you a taste of what 3D modeling is all about, should you be interested), head here (https://download.blender.org/release/Blender2.79/) to download **version 2.79** of Blender, a free and open-source 3D modeling toolkit that rivals most commercial equivalents. (If not sure

which file is right for your system, go ahead and download the newest version and check the filename of that file, then delete it and download the version 2.79 equivalent!) Using and mastering Blender is in and of itself a tremendous skill and art form and not required of this class, but do experiment if you feel so inclined! You can find some fantastic learning resources here (https://docs.blender.org/manual/en/dev/) and here (https://www.blender.org/support/tutorials/)! Should you wish to tinker with the models used in this project, you can find the helicopter, skyscrapers, and airplane (which I so crudely modeled) in the `Assets/Resources/Models` folder of the Unity project you've downloaded!

Downloading Unity

> Having some trouble with Unity? The staff has found that **Version 2018.4.28f1 (https://unity3d.com/unity/qa/lts-releases)** has worked well for them on a variety of different operating systems, if the below doesn't work for you.

You will of course need to download Unity before you can run the distro code and see your project, so do just follow the link here (https://unity3d.com/unity/beta) to download Unity's open beta. The setup is very straightforward, but you will need a Unity ID in order to use the software (which is free!), so do visit this link (https://id.unity.com) to create one; you should also be prompted to create a new Unity ID via the software's launcher once it's downloaded onto your computer.

Once you've downloaded and logged in to Unity, just click the "Open" button on the launcher and browse to the folder of the cloned code from the distro, and the project will open up!

But wait… nothing seems to load into the scene once you've opened it! With the project open in Unity, navigate to `Assets/Resources/Scenes`, and then select `Main` in the file browser at the bottom of the screen, double-clicking to open, and all should be loaded into the scene view!

Note: If you find that some of the models in your scene are not showing up, it's likely because you either don't have Blender installed yet (see instructions above), or you opened the project prior to the Blender installation. If you already have Blender installed and still don't see anything, do just right-click, in the Unity editor, any of the models located in `Assets/Resources/Models` and select the `Reimport All` option, which should fix missing models after a few moments of loading!

Next-Level

Welcome to your ninth assignment! Unity is a lot to take in at once, but beneath all of the details, we'll find that this set of tools will allow us to be our most flexible and productive yet, even when coding in C#! As such, this assignment is meant less to be intensive and more just to get a grasp on navigating Unity and understanding how things work.

Specification

- *Add Gems to the game that spawn in much the same way as Coins, though more rarely so. Gems should be worth 5 coins when collected and despawn when off the left edge of the screen.* We have all of the pieces for this already implemented in the `Coin` and `CoinSpawner` classes, so it should suffice simply to make some new classes for the `Gem` and `GemSpawner` behaviors! In the Proto resource pack included in the `Assets` folder, you'll find a model for a gem you can use, but feel free to import your own! You'll need to make a prefab, recall, that you can attach to the `GemSpawner` component, should you choose to implement it similarly to what's in the distro. There are of course other ways to implement this behavior, so feel free to experiment with the software as a chance to learn it all the more thoroughly if curious (but if you do decide to place it somewhere more unorthodox, make extremely sure when you commit your code that the staff is able to find it relatively quickly)! Do remember to make `Gems` worth 5 coins instead of just 1, and ensure they're more rare than `Coins` as well! Aside from that, they should behave identically to `Coins`, including moving automatically from right to left and despawning when past the left edge of the screen!
- *Fix the bug whereby the scroll speed of planes, coins, and buildings doesn't reset when the game is restarted via the space bar.* This one's a one-liner; note that static variables aren't actually reset upon loading a scene, so a place to check would be the `SkyscraperSpawner`, as the `speed` field therein is what actually drives the speed for `Skyscrapers`, `Airplanes`, and `Coins`! However, we won't find that this is the place where the game is reset upon pressing the space bar, and thus changing `speed` here doesn't make much sense; any guesses as to where the code for resetting the game could be located?

How to Submit

When you submit your project, the contents of your `games50/projects/2018/x/helicopter` 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/tree/games50/projects/2018/x/helicopter/Assets/Resources/Scripts` (where `USERNAME` is your own GitHub username as provided in the form, below) we should be brought to the `Scripts` subdirectory for Helicopter Game 3D. 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), 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), push your work to `https://github.com/me50/USERNAME.git`, where `USERNAME` is your GitHub username, on a branch called `games50/projects/2018/x/helicopter` or, if you've installed `submit50` (https://cs50.readthedocs.io/submit50/), execute

   ```
   submit50 games50/projects/2018/x/helicopter
   ```

   instead.
3. Record a screencast (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) (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:
     - Gem pickup
     - Speed reset
4. Submit this form (https://forms.cs50.io/0dbf9e2f-b121-4558-a419-0d09f49b5b98).

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

progress!