

pyGame Lecture #7

DESIGNING A LARGER GAME IN PYGAME

I. Time to Brainstorm

Before you can start coding your game, you need to know what you want your game to be. Therefore, you're going to need to come up with some ideas. During this brainstorming stage, you'll write down all of the game ideas that you might want to create. If you're having trouble coming up with things off the top of your head, maybe think back on games that you personally might have enjoyed before. Of course, you won't be copying any of them, but they can provide some great inspiration. You can also look over some of the things you've worked on in this class, but keep in mind you'll want to differentiate from your earlier projects.

II. Deciding on an Idea

Now that you have a list of ideas, you'll want to pick one to make the focus of your project. There are a few things you should keep in mind. First, make sure you consider the time frame you have to complete the game. Even a simple flash browser game you might have seen online could have taken years to complete. Given the limited time you have, it's a good to pick an idea that's small and focused. You'll also want to make sure your choice is practical. What are you capable of with your current skill set? What game genres would be easiest to make with the tools you have in PyGame? What sounds like the most fun idea to you? It'll be easier for you to focus on and engage in an idea that captures your interest. This makes it more likely that the game will be completed.

III. Planning your Game

Next, you'll want to make a game plan (pun intended). Plan out everything you're going to add to your game from the top down. This means you'll start with the broader elements, which could include overall story, game structure, etc. Then, you'll narrow your focus to the small details, such as enemy types, level design, etc. When you're

making these plans, again make sure to keep in mind how much time you have. Also realize you won't just be coding; you'll be designing sprites, testing, debugging, and dealing with unforeseen obstacles. Planning is important to maintaining focus and also preventing feature creep, an idea that will be expanded on further down.

IV. The Programming

Now, you're finally ready to start actually coding your game! This project will be bigger than any other you've completed in this class, so implementing good coding practices is more important than ever. Keep separate files for different groups of code/functions/classes. This will make it much easier to find the elements that you need when you must review or make changes to old code. See the earlier lecture notes for help with this. Similarly helpful is keeping your code well-commented. Leave comments for classes and functions explaining their purpose or any other important information. You also might want to make regular backups of your code. Just keep a folder to keep. If something goes disastrously wrong, you can back up to an earlier functional version of your code.

Make sure to keep to your plan! Feature creep, defined as the tendency for elements involved in a project to increase as the project goes on, is a dangerous force. If any interesting ideas pop in your head while you're coding, write them down to come back to later. Cool or experimental ideas always come *after* the main functionality of your game is reliably implemented.