Systems Dev 2: Event-Driven Design

Jo Husack's Writeup!

Describe the design and event flow of the system:

There are a few things missing from what I designed in the other class, but i kept a good amount of things key to design

1. Players spawn in and slowly lose health over time, prompting them to find healing from the environment. This health drain increases with time.
2. Players can pick up items from the environment, which they can use to interact with the environment
3. Players will use their item pickups to destroy enemies (in this case, destroyables,) which drop things on destruction
4. Players will eventually either die of health loss, or destroy all the destroyables. In design, enemies would spawn procedurally, but we're short on time, so when all of the non-moving destroyables are dead, it's game over.

Explain how events improve modularity and gameplay responsiveness.

With events, you can easily get any actor to give a signal, knowing that you'll have another script listening for it. Afterwards, you can remember you have the other actor giving a signal, and add any amount of listeners to react upon hearing the signal.

These events let you program one system at a time, instead of wiring everything up to a single source.

Reflect on challenges and solutions during development.

One challenge I faced was scope. Scoped back, cut the enemies with AI out of the picture, good job there.

Another major challenge was getting the inventory system to then have an "equipped item." I ended up going with a 2nd script that read from my inventory, but for a while I was attempting to change the existing dictionary, add a 2nd dictionary, and eventually cut all that out of the picture.

Lastly, I only used event systems for the inventory system! That wasn’t a problem, but I wanted to show I knew how to use events. I ended up adding a win condition, and listening to a signal from my destroyables/enemies.