CS 45: Operating Systems

Lab 5: Event Processing

DUE November 21st at 11:59 PM

We're at the point in the semester where we need some extra fun.

So you are going to create a little game (about OS) using `raylib` and then describe how that game (& `raylib`) performs event processing using `poll`, `select` or something else. Remember different devices (i.e., peripherals like the mouse, keyboard, gamepad) are generating events, graphics are being updated, concurrently—all the time—how do we deal with this?

Create a `COLLAB.md` file to keep track of any outside resources you might use.

1 Get `raylib` going

First we need to grab raylib and build it.

Let's put `raylib` outside this github repo (we don't want to check in the entire thing).

```
$ cd SOMEWHERE_ELSE
$ git clone https://github.com/raysan5/raylib.git raylib
$ cd raylib
$ mkdir build && cd build
$ cmake -DBUILD_SHARED_LIBS=ON ..
$ make
```

Then we can try some examples:

```
$ examples/core_input_keys   #core_2d_camera  core_2d_camera_platformer
```

Then we can modify it a little, and verify it did indeed change:

```
$ code ../examples/core/core_input_keys.c # change the program
$ make
$ examples/core_input_keys
```

2 How does `raylib` process events?

Describe how `raylib` processes events like gamepad button presses and keyboard smashes here. Talk about the **SDL core** we are using, but also one other, for example, `Android`. Think back to that "Tearing apart `printf`" article we read. Really try to understand what's going on from button smash to pixel colored.

3 Make an OS Game!

Create a game that is inspired by an Operating Systems topic, and specifically a topic talked about in Downey's **Little Book of Semaphores**: e.g., Dining Philosophers, Deadlock, Barriers, Child Care, Room Party„ etc.—lots of great ideas in there, be creative! The game is just **inspired by a semaphore topics**, you don't have to necessarily use semaphores, try to integrate the ideas into the narrative or the mechanics.

3.1 Game Concept

Describe your game concept and summarize how semaphores are used to solve that concurrency problem.

3.2 Implementation

Describe your implementation of the game and add your source file to the raylib separate template repo.