INTERRUPTS, POLLING, AND EVENTS

POLLING

- It is a while loop – keeps checking condition
  - If it is a lifetime loop, then the condition is 1
- Inefficient, poor design
- Wastes CPU time and energy
- Simple design

INTERRUPT

- Invokes a function when a physical input occurs
- Harder to implement
- More expensive
- No wasted CPU time, no wasted energy (in theory)

EVENTS: CAN BE TRIGGERED FROM INTERRUPT OR POLLING

- Some condition occurred and the system or library invoked a function
- i.e., TouchesBegan – iOS event triggered when the user touches anywhere on the screen. It passes info that contains where, when, how many fingers, etc.
- This is triggered by an event and is defined in software, not hardware
  - Although, hardware can trigger the event to occur
    - A lower-level concept, so I will not go into detail

EVENT POLLING EX

```
while (true) {
	switch(event) {
	case MOUSE_MOVED:
		mouseMoved();
		break;
	case MOUSE_CLICK:
		handleClick();
		break;
	default: break;
	}
}
```

INTERRUPT EX

```
void IRQ_Handler(void* addr) { /* do stuff */ }

/// Begin main
// Setup Interupts
Xil_ExceptionRegisterHandler(5, IRQ_Handler, (void*) &sw );
// More config

while (1) {
      ;
}
```

INHERITANCE EXAMPLE IN SWIFT

- More clearly shows the purpose of abstract classes