My envelope is essentially a finite state machine consisting of 5 states. Attack, Decay, Sustain, Release and IDLE. The times for each state (apart from IDLE and Sustain) are converted from milliseconds to samples. I use a tick method to count the samples and when the amount of samples reach the state's threshold it is automatically set to the next stage. Getting passed sustain and going from IDLE to attack require user input. For my envelope I made use Martin Finke's ADSR calculations (link to be provided within the code). These calculations generate an exponential curve from the current amplitude to the state's desired output. This same envelope is placed on the oscillator pitch.

The observer design pattern is a "one to many" subscription where several objects rely on notifications from a subject object. Observer objects can subscribe, or attach, to one observable object and use its notifications to further their own processes. In the case of this project I used a tick() method (which served as notify) to synchronise all the different tick() methods in the observer objects. In essence a clock syncing protocol.