* Have the user instantiate gesture processor in program
  + gp = GestureProcessor([gestureFile])
  + If no file specified (or invalid file), basic gestures loaded
* On initialization, bind actions to available gestures
  + gp.bind(gestureName, fn)
* In continuous loop, have the GestureProcessor take one step (frame) forward
  + gp.process()
  + Will call bound functions whenever they happen
  + Nonblocking, so that it can run in animation thread
* Record additional gestures by saving the next gesture that is detected
  + gp.record()
  + Will set another variable with the last gesture detected, since not necessary that the gesture will be detected right at the call of record. Can't have blocking.
  + Programmer can then delete the new gesture as necessary
* System will always be learning how the user performs gestures
  + No need to have a dedicated train() method
* See sample Tkinter and EventBasedAnimationClass code for example