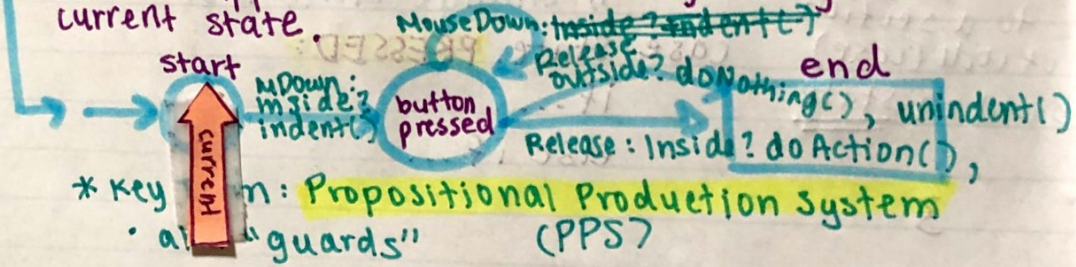
Topic: State machines with EssentialGeometry

myAndroid App

① Background: what exactly are s. machines and essential geometry?

a) **State machines:** basically a **flowchart** outlining what an application should do regarding the current state.



* Key m: **Propositional Production System** (PPS)
  - al "guards"
  - ex: MouseDown: InsideAboveThumb? scrollUp()
    (MouseDown: event; InsideAboveThumb?: condition; scrollUp(): action)

b) **Essential Geometry:** says **what part** of your view is **actually clickable/actionable** (views are default rectangular)

→ only want the user clicking w/in circle to change anything.
- give a name to each part of the view:
  - ex: InsideButton, OutsideButton
- **calculate user touch vs. circle**, ex: distance formula from circle center to touch

② outline of steps:
1. App starts, nothing happened yet → state = START
2. User presses down **inside button circle** → state = PRESSED, indent()
   - 2a. User drags finger **inside circle** ↔
   - 2b. User drags **out of circle**
3. If user releases **in circle**, doAction()
   else: doNothing()
4. un-indent()

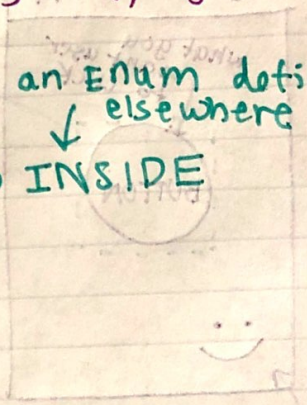
Code → in the View → onTouch

```
public boolean onTouch(MotionEvent e) {
    geometry = essentialGeometry(new PointL(e.getX(), e.getY()));
    switch (state) {
        case state.START:
            // check if geometry is equal to INSIDE
            // change state if so
        case state.PRESSED:
            if ...
            else if ...
```

this is given to you in your HW!

an Enum defined elsewhere

makes it EZ to jump to a section of code (case) based on state



3