CS 173 Lecture 21a: State Diagrams modeling Programs

Special Directed Graphs

vertices are "states": relevant description/variables of the program at a given step/time

edges are "transitions": program moving between states.

```
Euclid GCD (a,b):
1.  x ← a
2.  y ← b
3.  while (y>0):
4.      r ← remainder (x, y)      // x = yq + r   0 ≤ r < y
5.      x ← y
6.      y ← r
7.  return x
```

States: (x, y) i.e. program is in state $(252, 198)$ iff $x=252$, $y=198$

transitions: $(n, m) \xrightarrow{4\text{-}6} (m, \text{remainder}(n, m))$

$(252,198) \to (198,54) \to (54,36) \to (36,18) \to (18,0)$

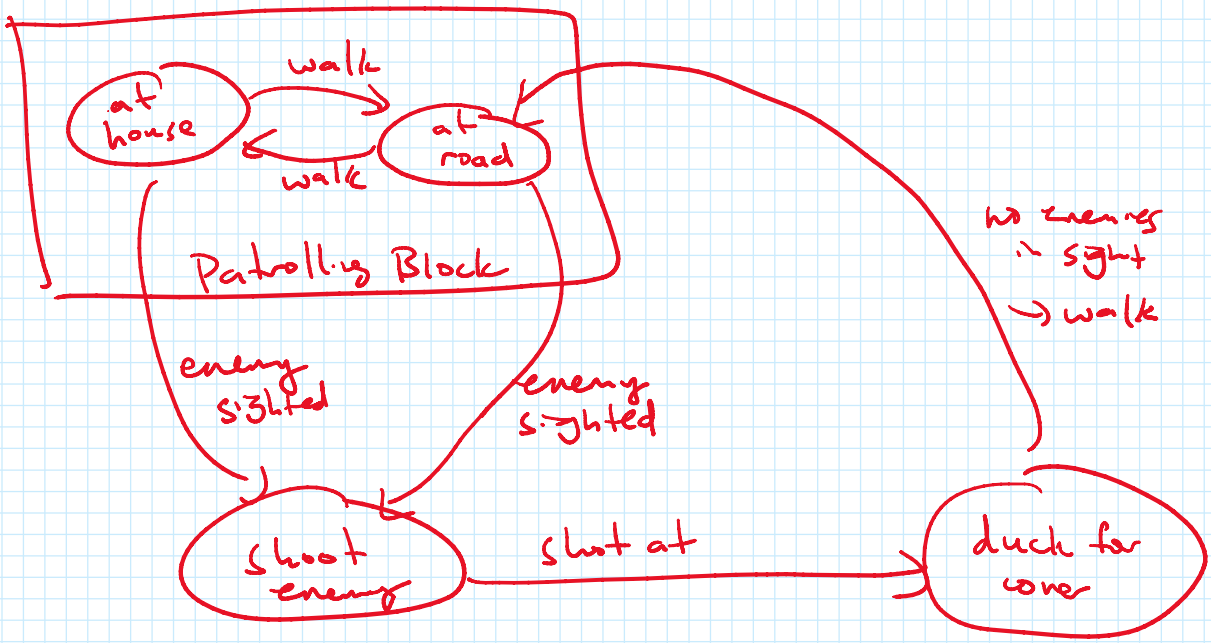
$\cdots \to (54,18)$

Given a program: state diagram gives a way to visualize / represent control flow

Also: start w/ a state diagram → turn that into code.

State diagrams can be vague/abstract

e.g. World's Dumbest AI.



↳ later: converted these diagrams to code.

less dumb AI