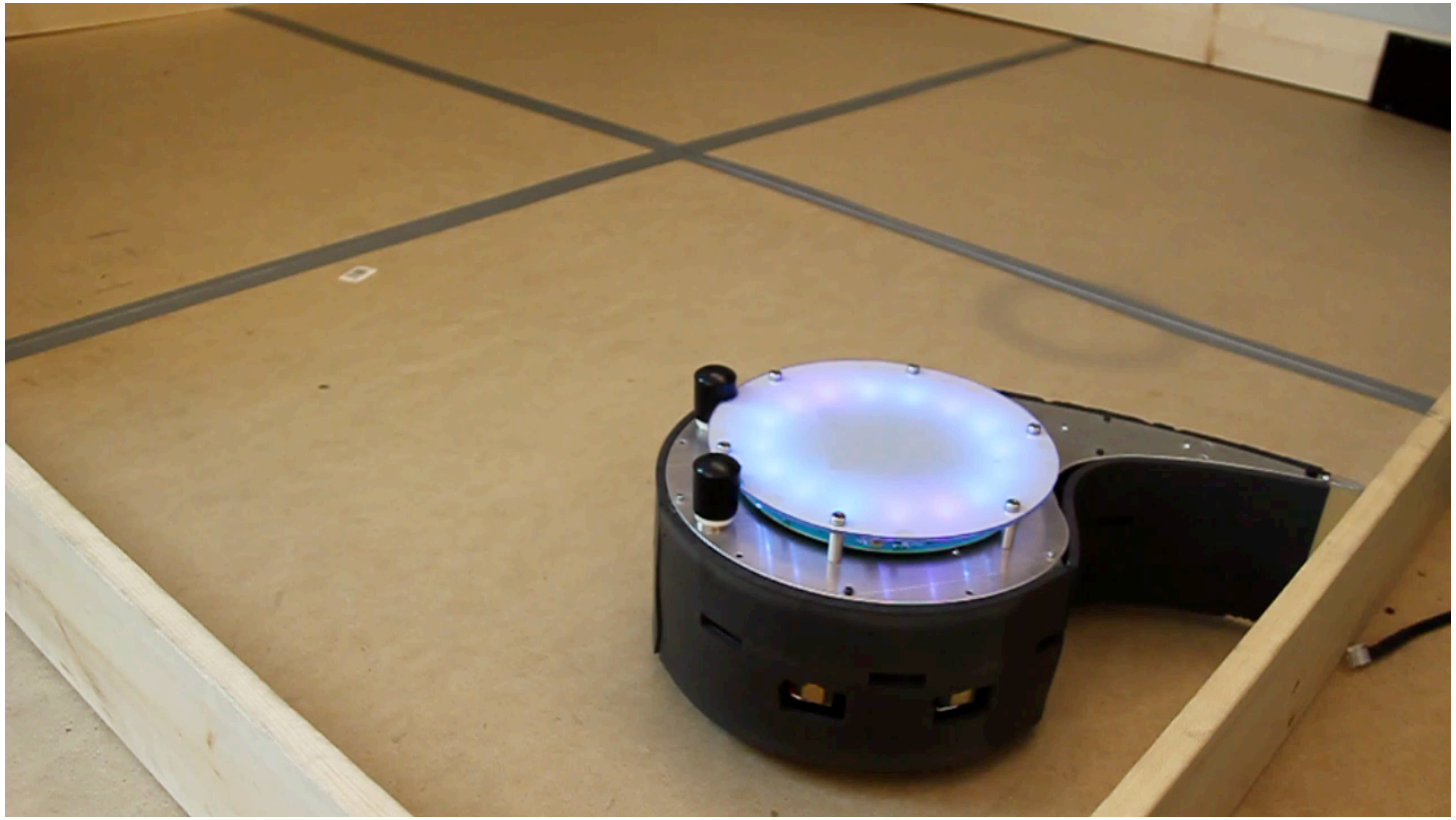
Markov Decision Processes

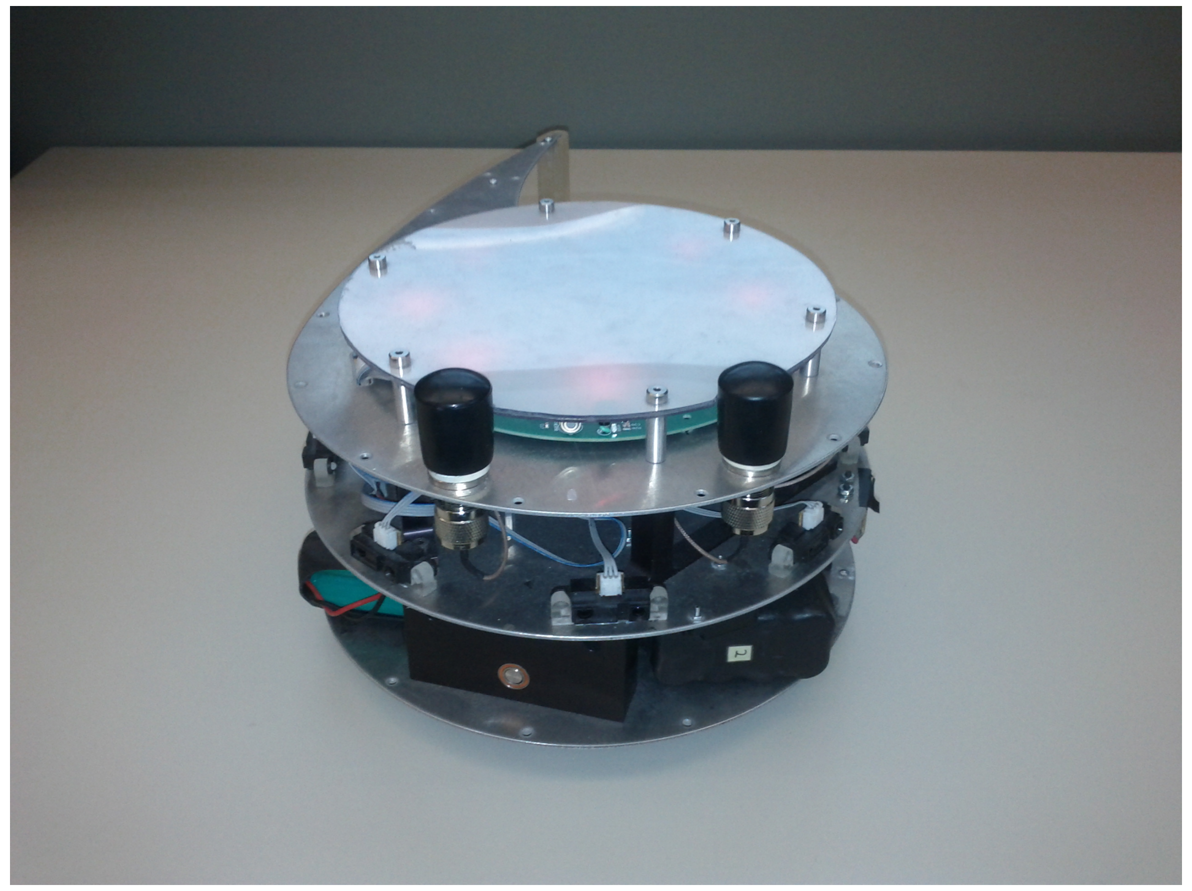
CMPUT 655
Fall 2022

Example MDPs

- Consider this robot called the critterbot

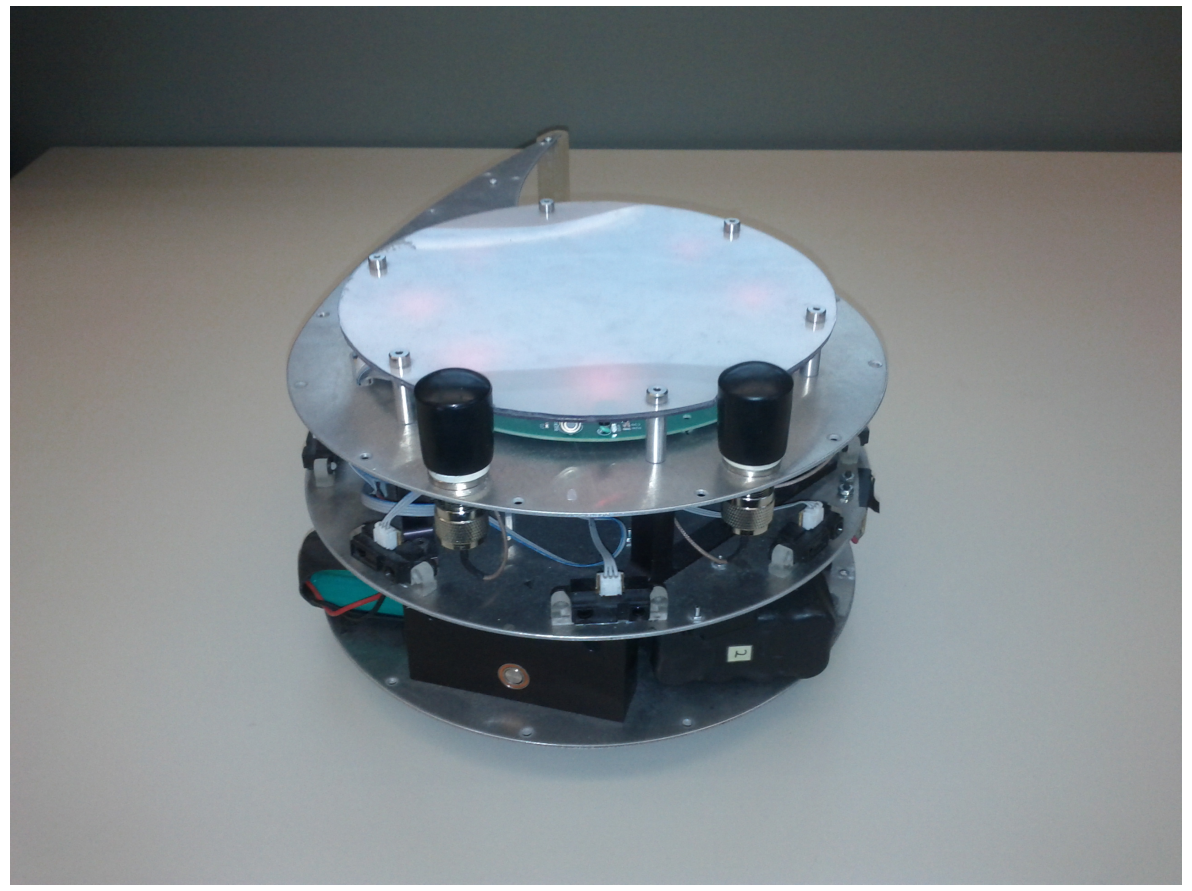


Robot sensors

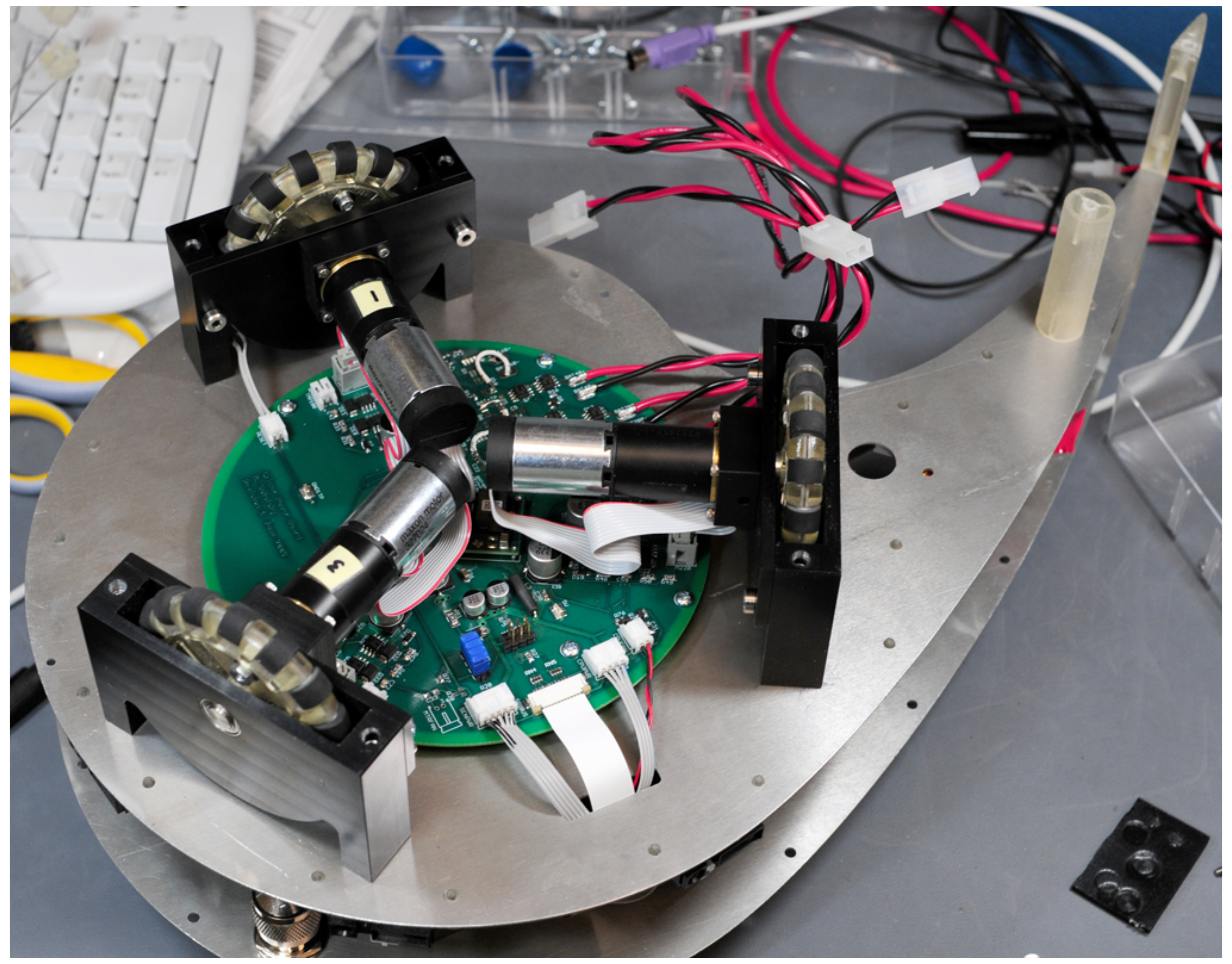


- This robot has **sensors** all over it:
  - Distance sensors
  - Light sensors
  - Thermal sensors
  - Motor information like speed, current, velocity, and temp
  - Accel XYZ
  - Rotational vel
  - etc

Robot actuation



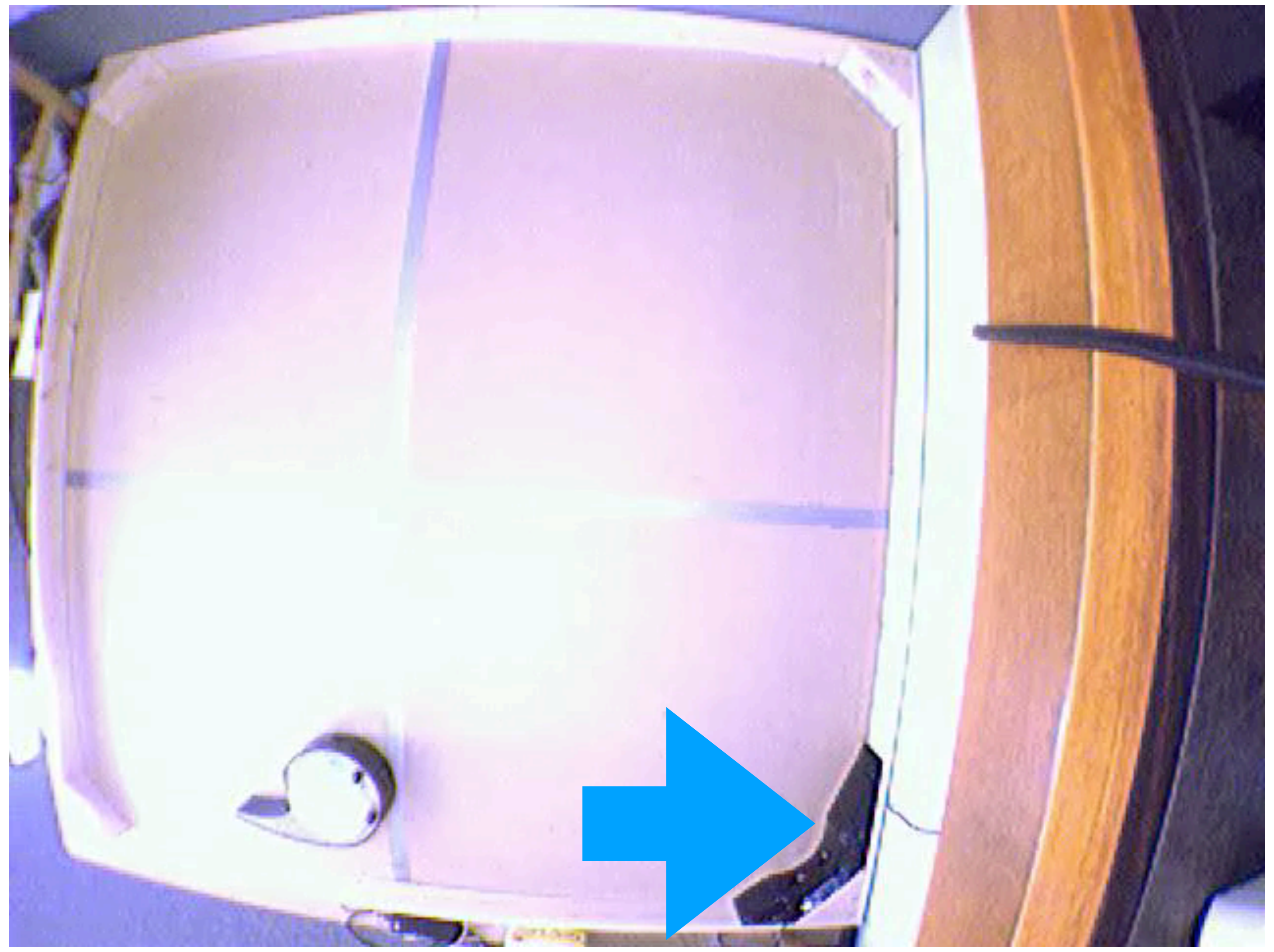
- This robot has an omni directional drive system:
  - Can move in any direction on a surface



An RL robot MDP

- Imagine we wanted this robot to **goto the light quickly**
  - What is the **state**?
  - What are the **actions**?
  - What is the **reward**?
- In other words how to formulate this as an **RL problem**?

Robot state



- Perhaps the sensors are so good that every situation in the pen looks **unique** to the agent:
  - The agent would not do better by **remembering** a history of the sensors
  - This is true because of the **IR beacon on the charger**, and **magnetic** sensors + all the other sensors
- We could also use an overhead camera + a localization algorithm to extract X,Y, theta position —Markov state
- It would be easy to imagine that if the robot only had distance sensors and the pen was square, that the state would not be fully observable

Robot actions

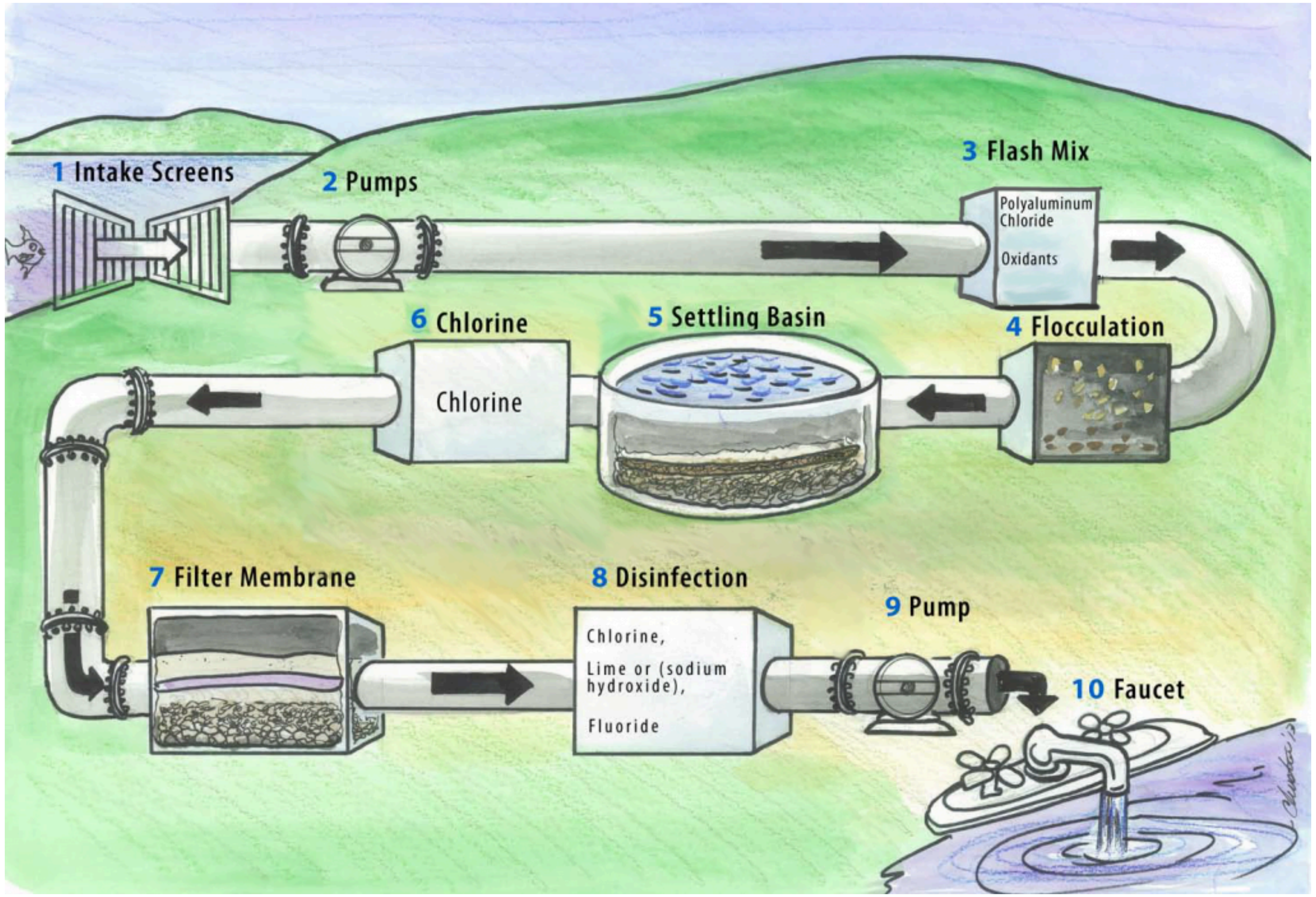
- This robot can be controlled in two basic ways:
  - X,Y,theta mode -> specify amount of translation in X and Y, and rotation
  - Voltage mode -> how much voltage to send to each of the three motors

Robot Reward

- To encourage light seeking
  - -1 per step, until front light sensor > threshold
  - OR
  - Reward = front_light_sensor_reading
- In both cases terminate when light sensor > threshold; **Episodic problem**
  - What is gamma?
- How do we reset to the state state? ME :(

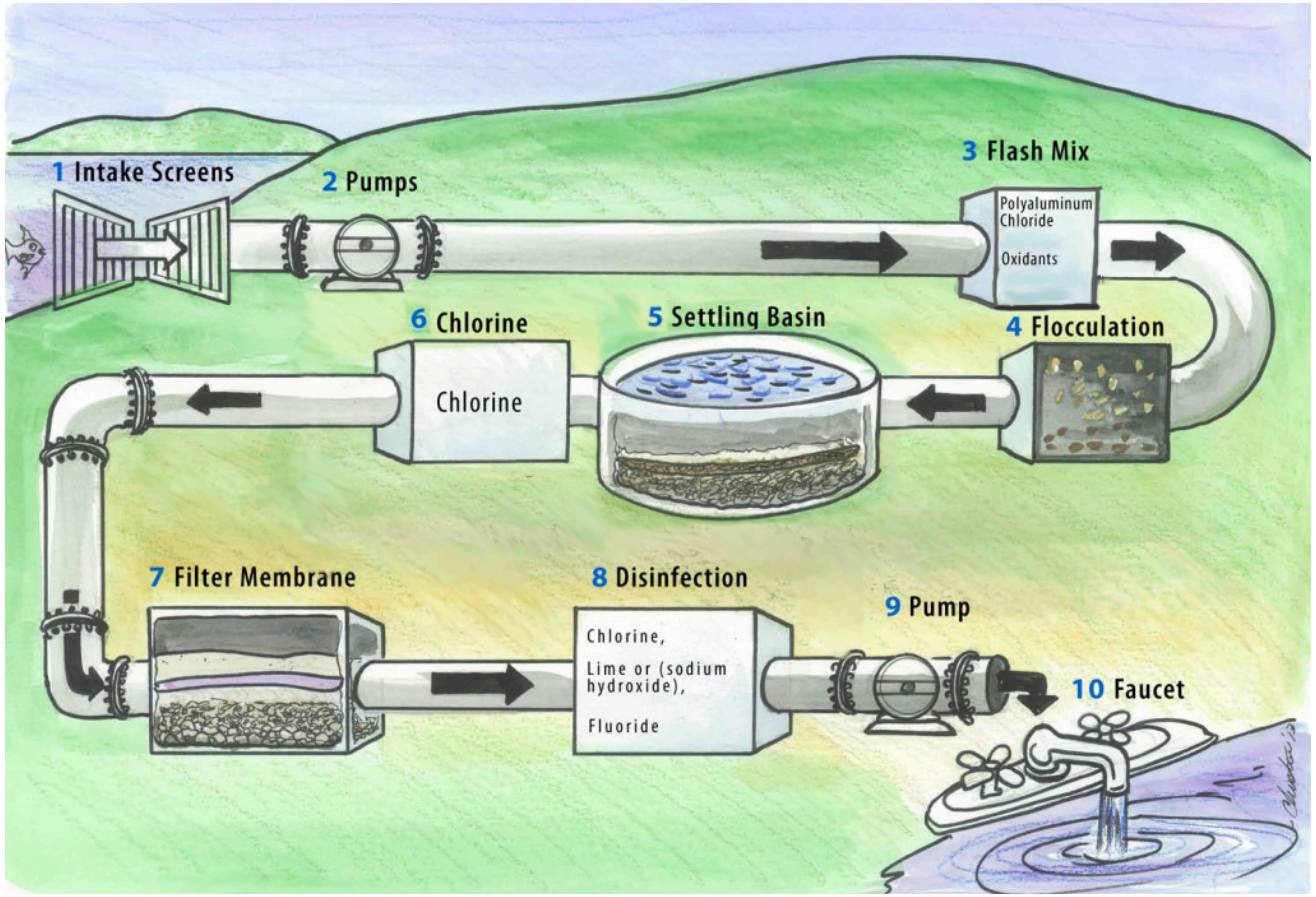
Fresh Water Treatment

- Water from the river
- Passes through many stages:
  - Chemical treatment
  - Settling
  - Mixing
  - Filters
  - Then to you to drink



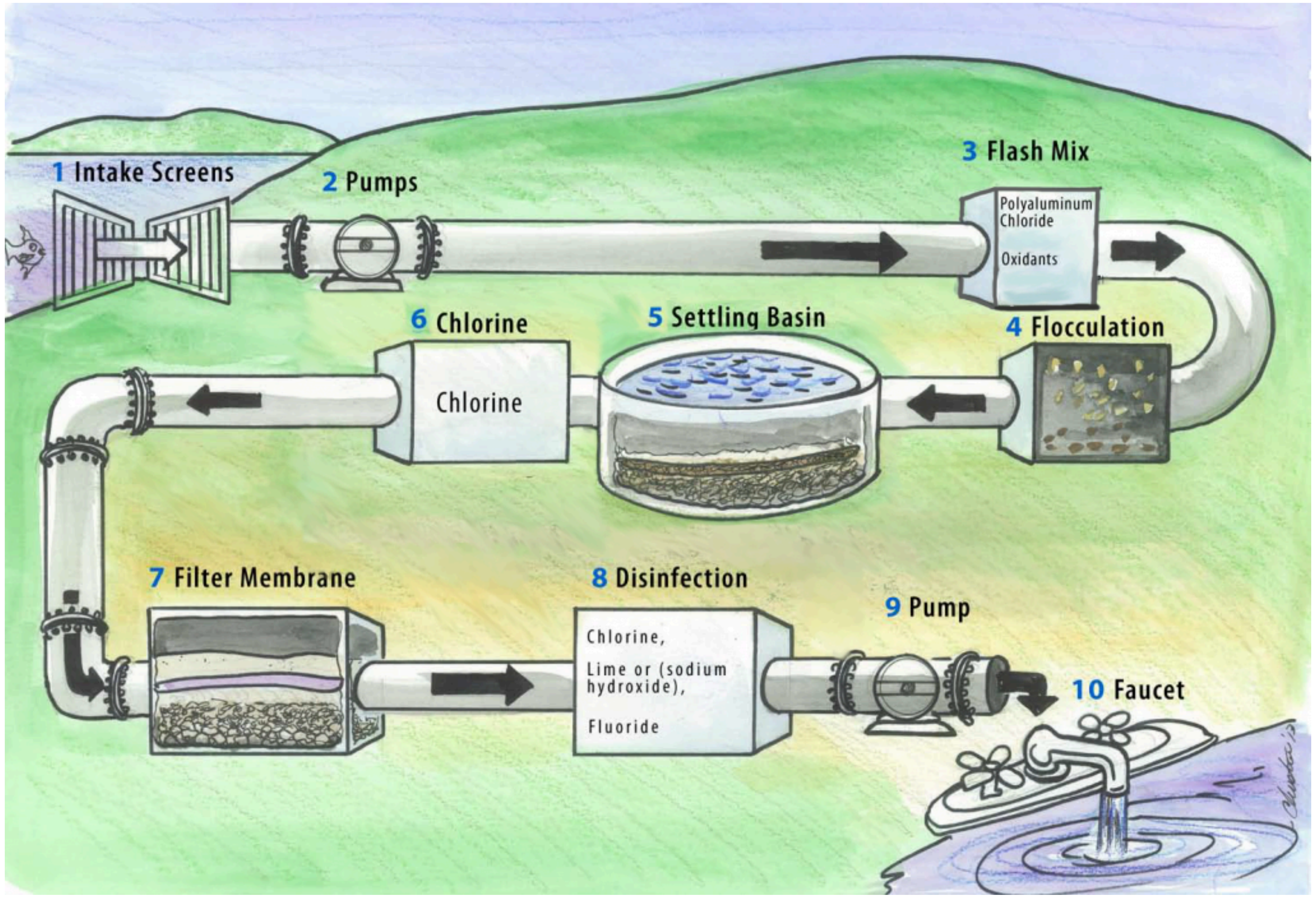
Fresh Water Treatment

- The reward is easy
  - R_t = Water flow - energy
- The system is **safe** by design:
  - Hard to break the components
  - Plant cannot produce unclean water
- Good for trial and error learning



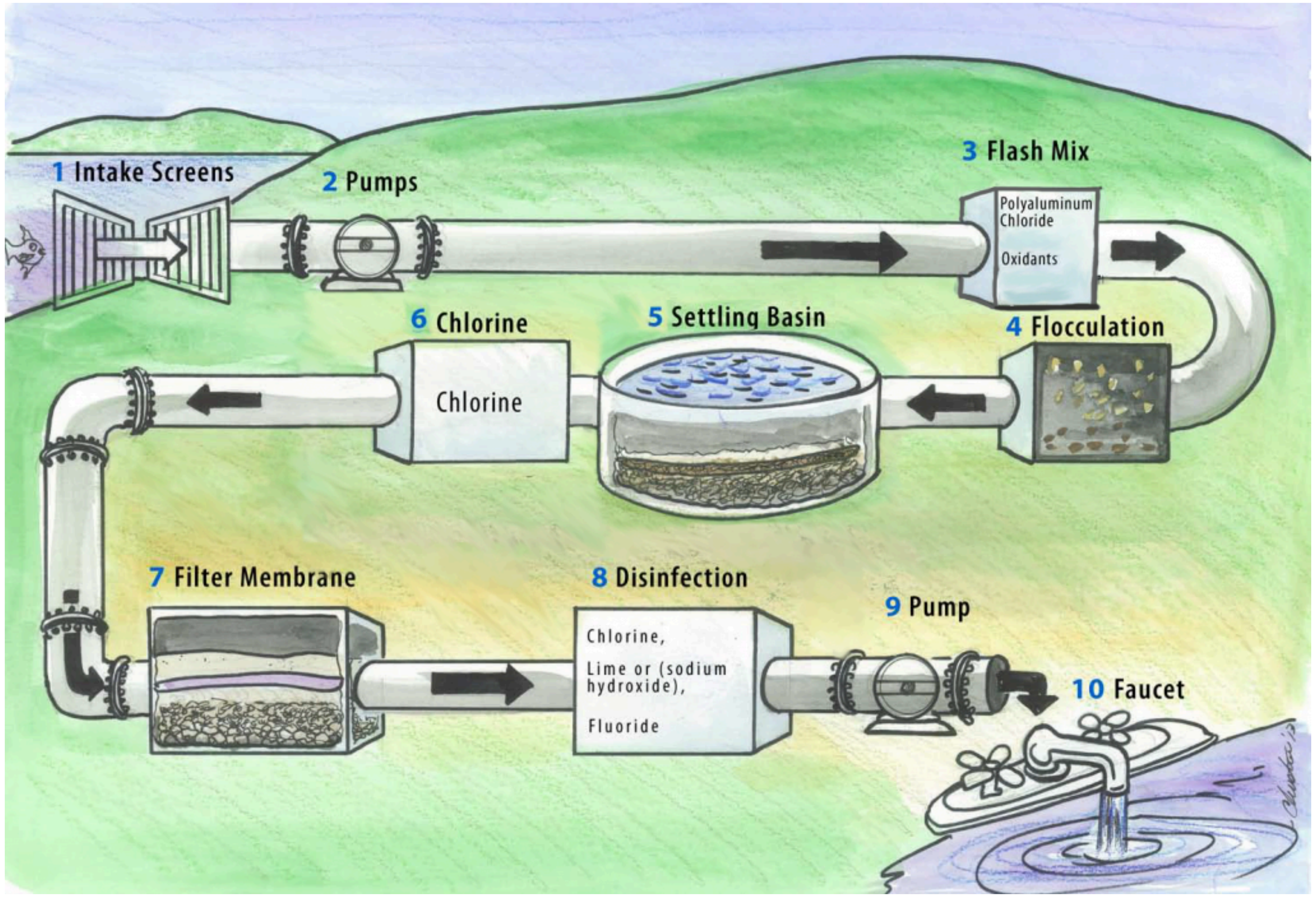
Fresh Water Treatment

- **Many sensors:**
  - Pump speeds and temp
  - TMP & other water data
  - Turbidity
  - Weather & future weather
  - River conditions
  - State of the filter
- Could be very close to **Markov state**



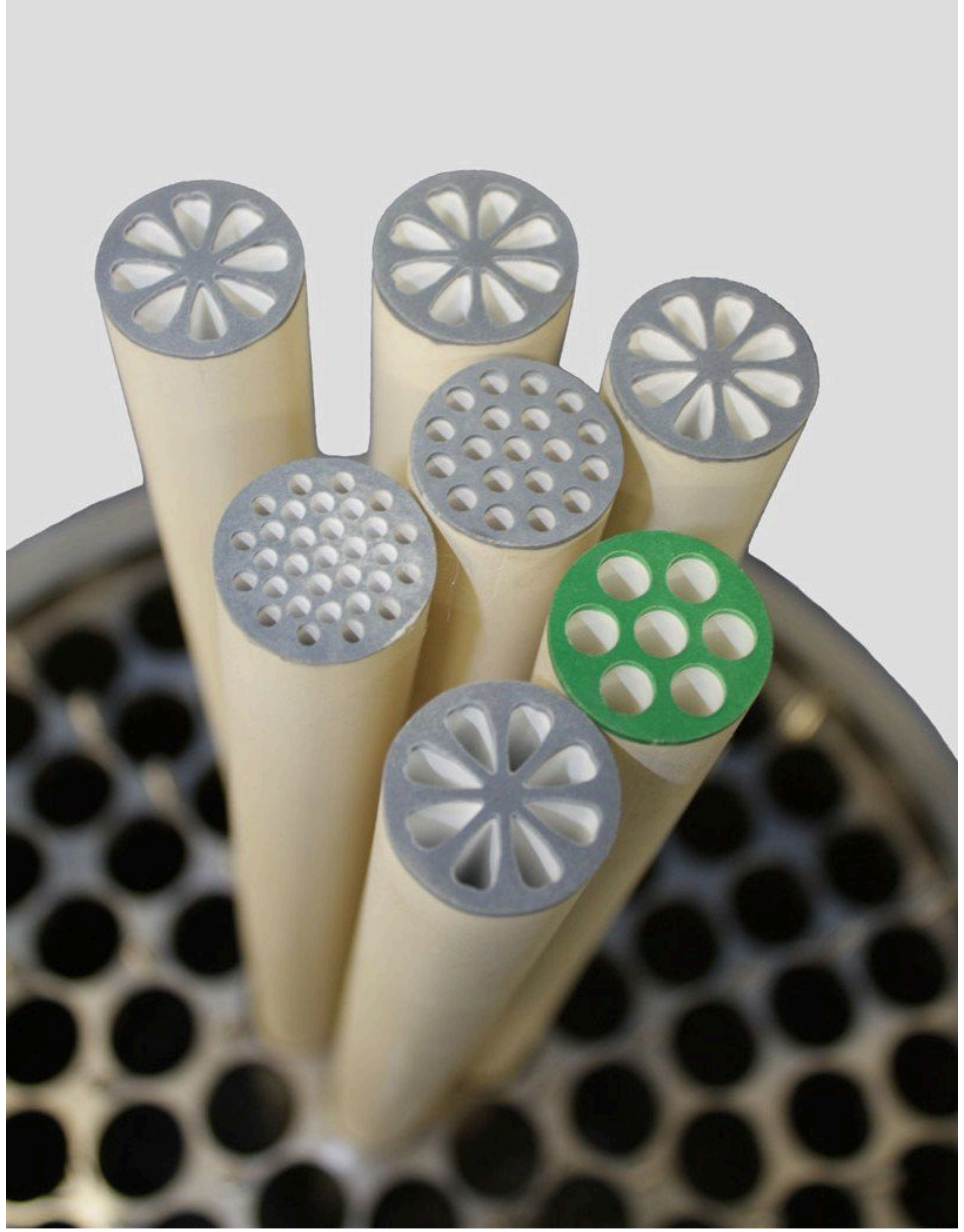
Fresh Water Treatment

- Many possible **actions**:
  - Pump speeds
  - Chemical treatments
  - **Backwashing the filter …**



Fresh Water Treatment

- The filter is comprised of hundreds of filter tubes
- They get full of dirt and other stuff
- To clean them we just run the system backwards (**backwashing**)
    - But backwashing uses a **lot of energy** and **stops water production**
- **Action**: backwash or not on every step



More example MDPs

- White in 1969 (not me!!)
  - Salmon Harvesting
  - Agriculture: how much to plant based on weather and soil state.
  - Water resources: keep the correct water level at reservoirs.
  - Inspection, maintenance and repair: when to replace/inspect based on age, condition, etc.
  - Purchase and production: how much to produce based on demand.
  - Queues

More example MDPs

- Flying REAL (small) Helicopters ~Andrew Ng et al

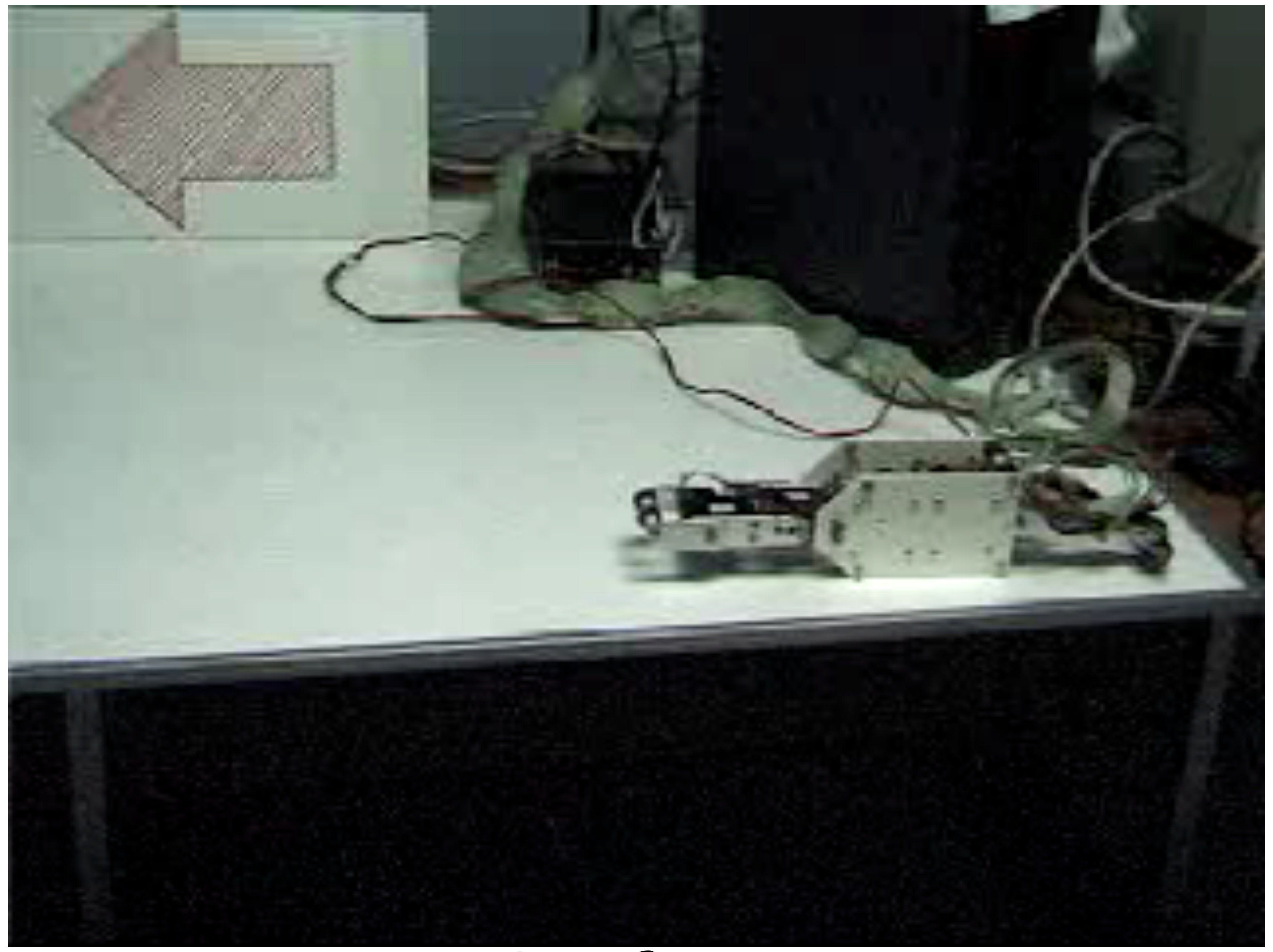
Heli MDP

- **Actions (multi-dimensional & continuous!!):**
  - Longitudinal and latitudinal Pitch controls
  - Main rotor pitch
  - Tail rotor pitch
  - Throttle

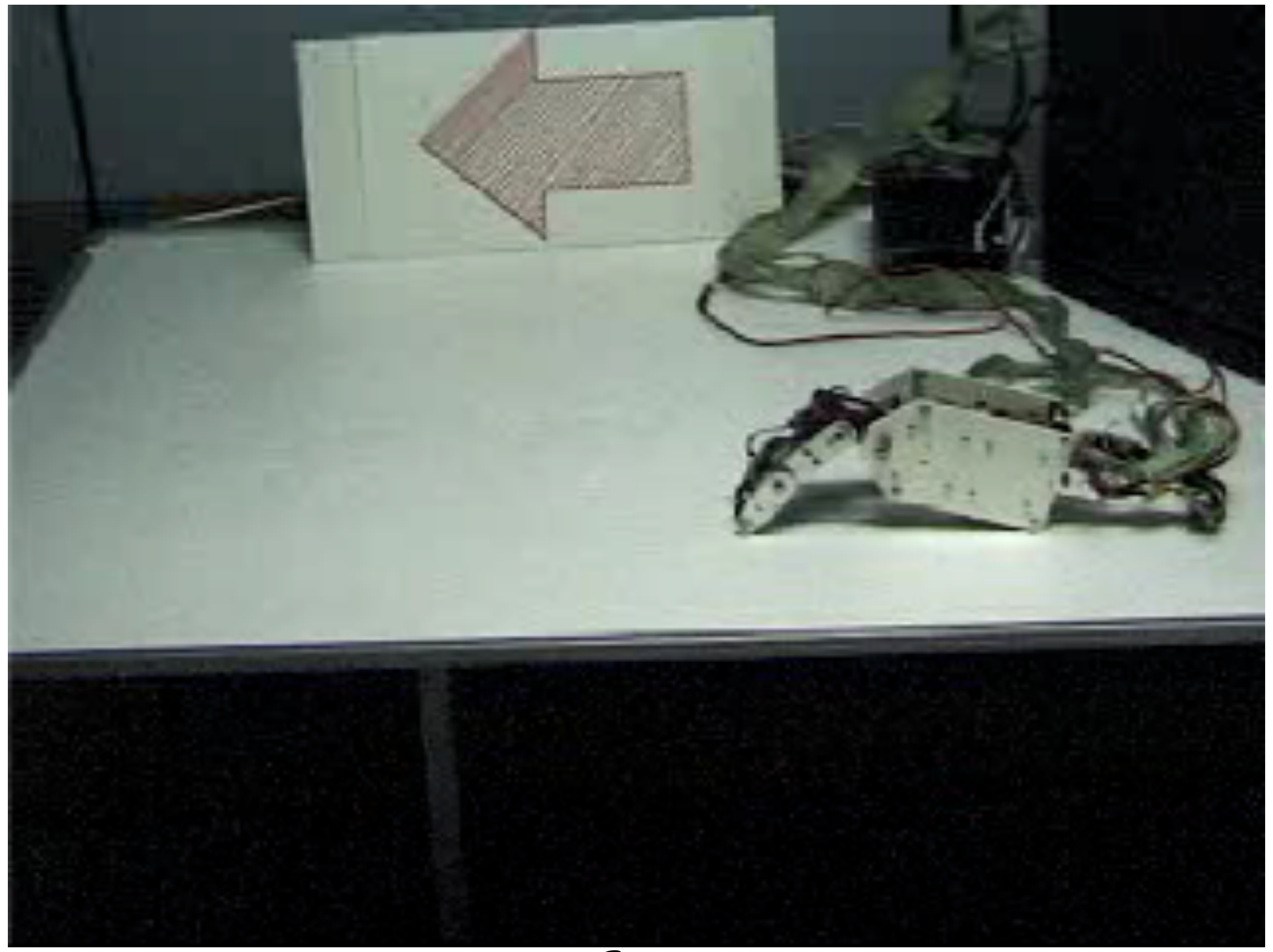
$$R(s^s) = -(\alpha_x(x - x^*)^2 + \alpha_y(y - y^*)^2 + \alpha_z(z - z^*)^2 \\ + \alpha_{\dot{x}}\dot{x}^2 + \alpha_{\dot{y}}\dot{y}^2 + \alpha_{\dot{z}}\dot{z}^2 + \alpha_\omega(\omega - \omega^*)^2),$$

- **State (8 dim):**
  - helicopter's position (x, y, z), orientation (roll ϕ, pitch θ, yaw ω), velocity ($\dot{x}$,$\dot{y}$,$\dot{z}$) and angular velocities ($\dot{\phi}$,$\dot{\theta}$,$\dot{\omega}$

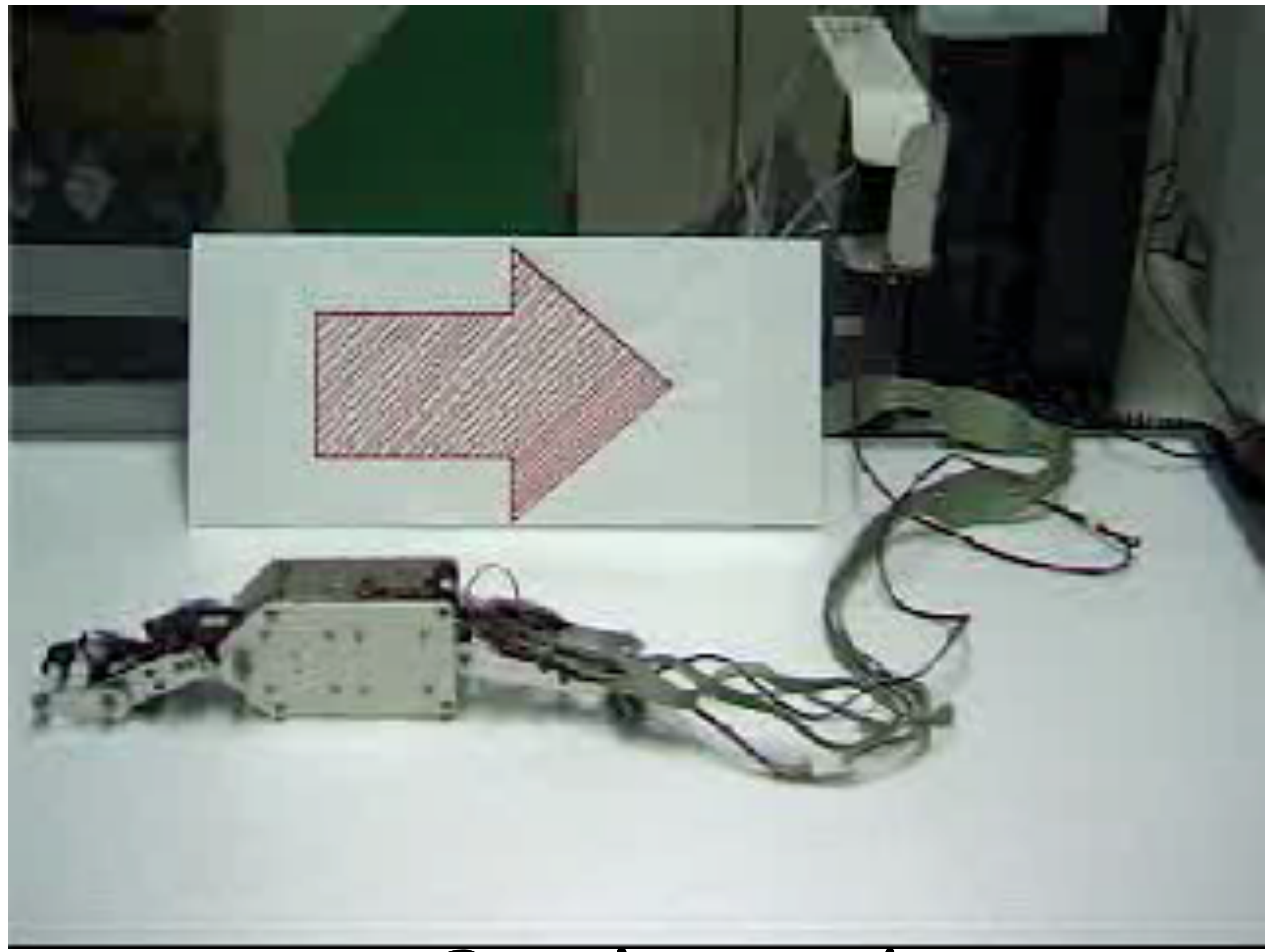
Hajime Kimura's RL Robots



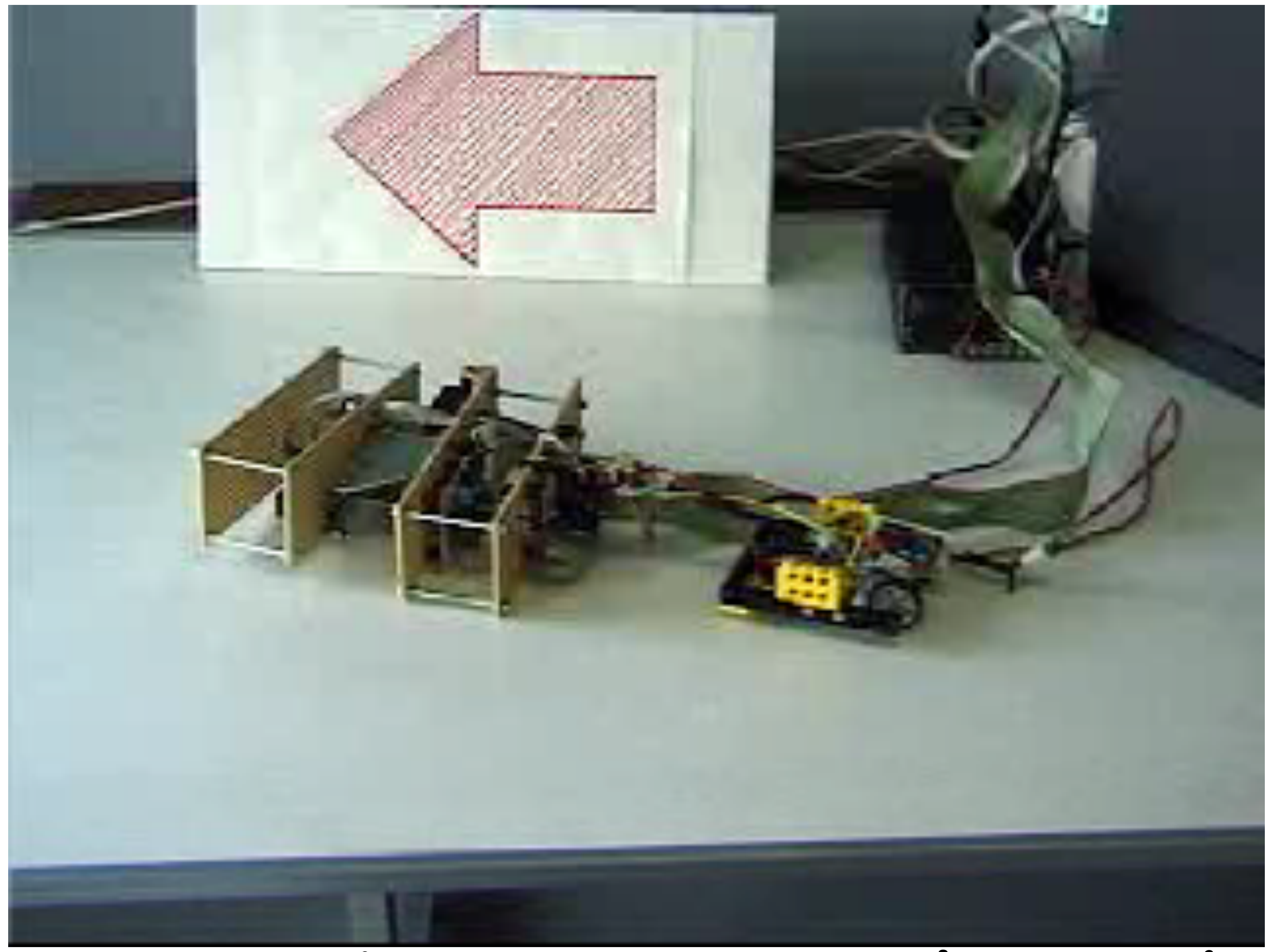
Before



After

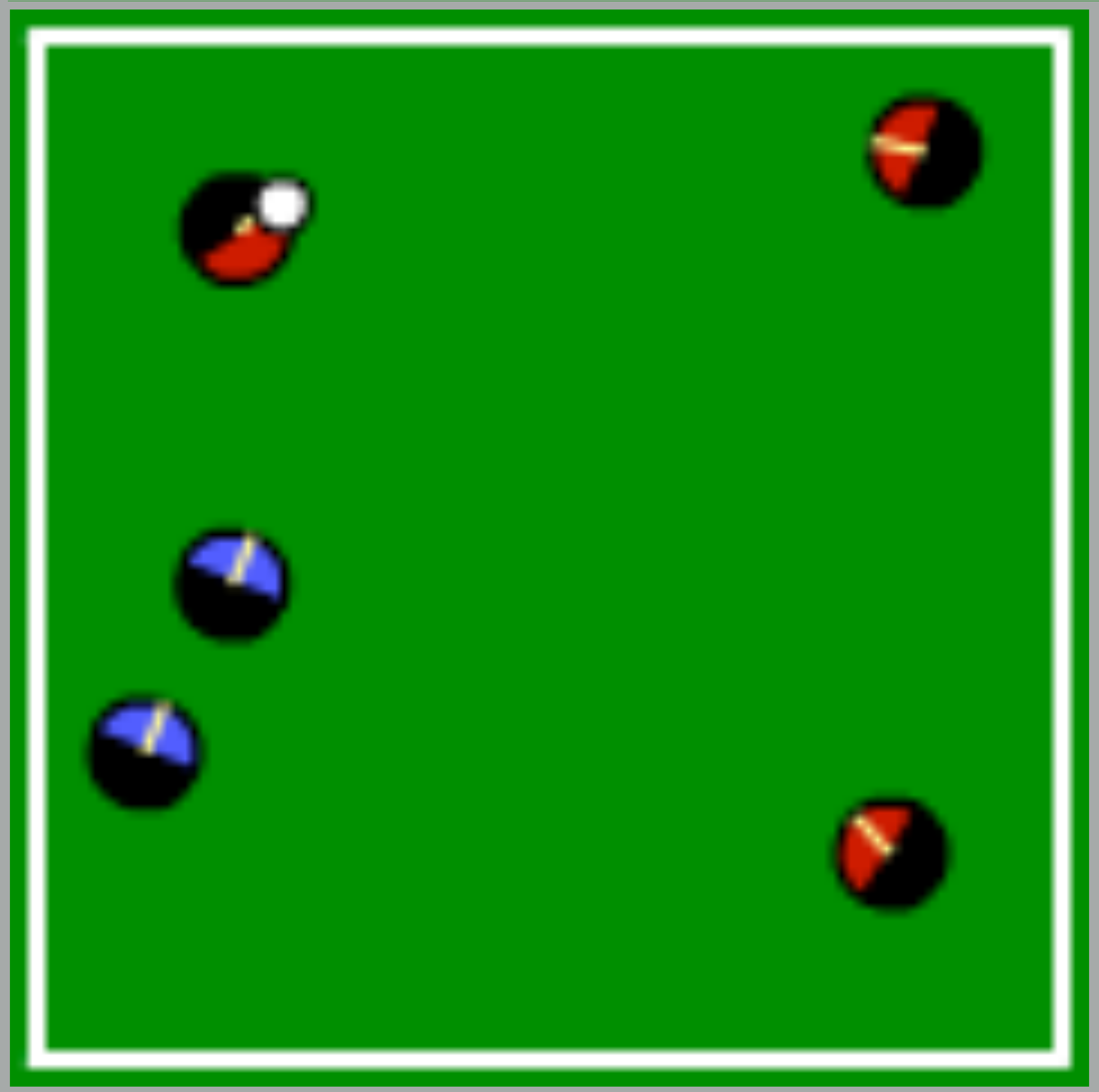


Backward



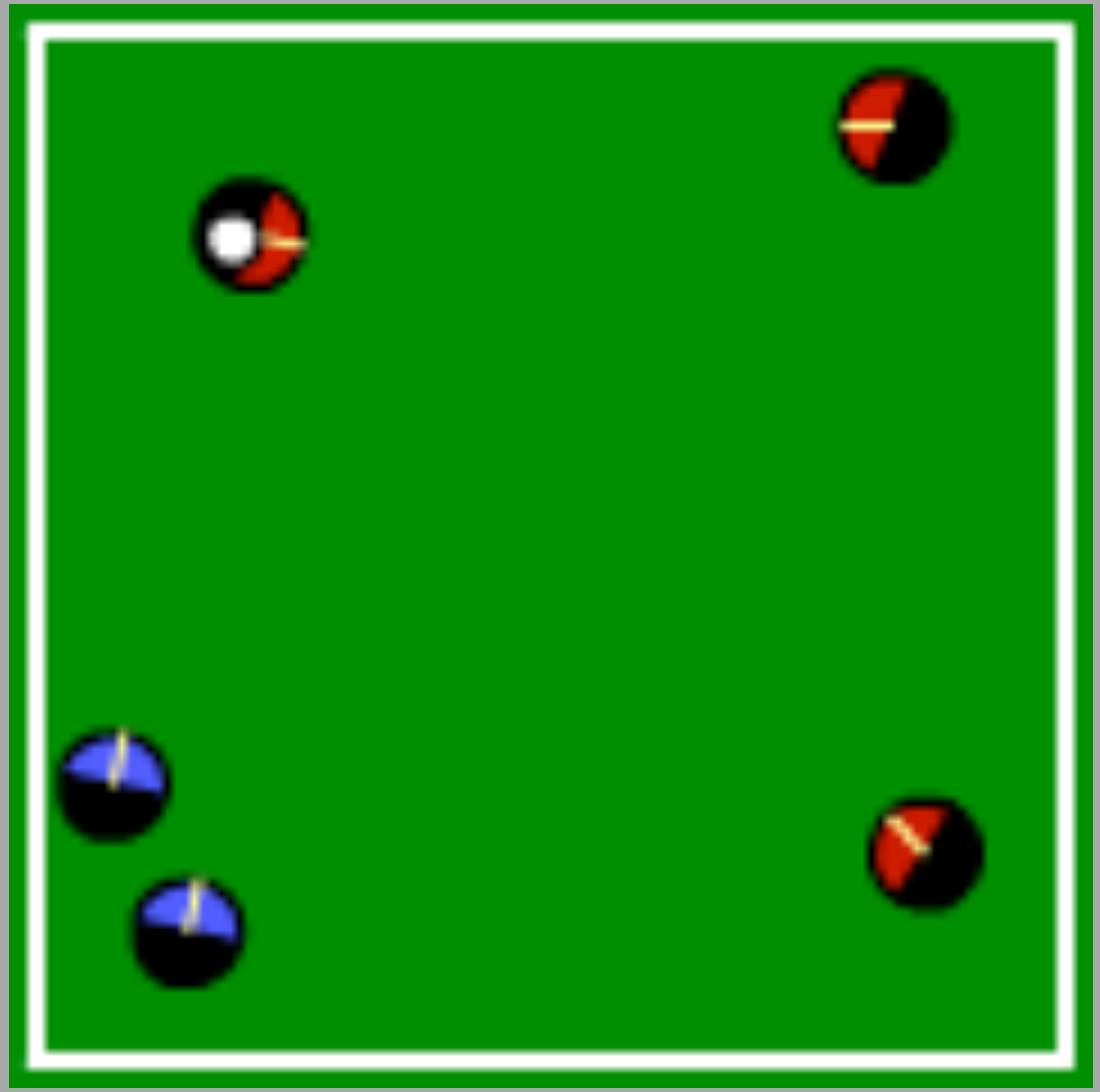
New Robot, Same algorithm

- Stone & Sutton

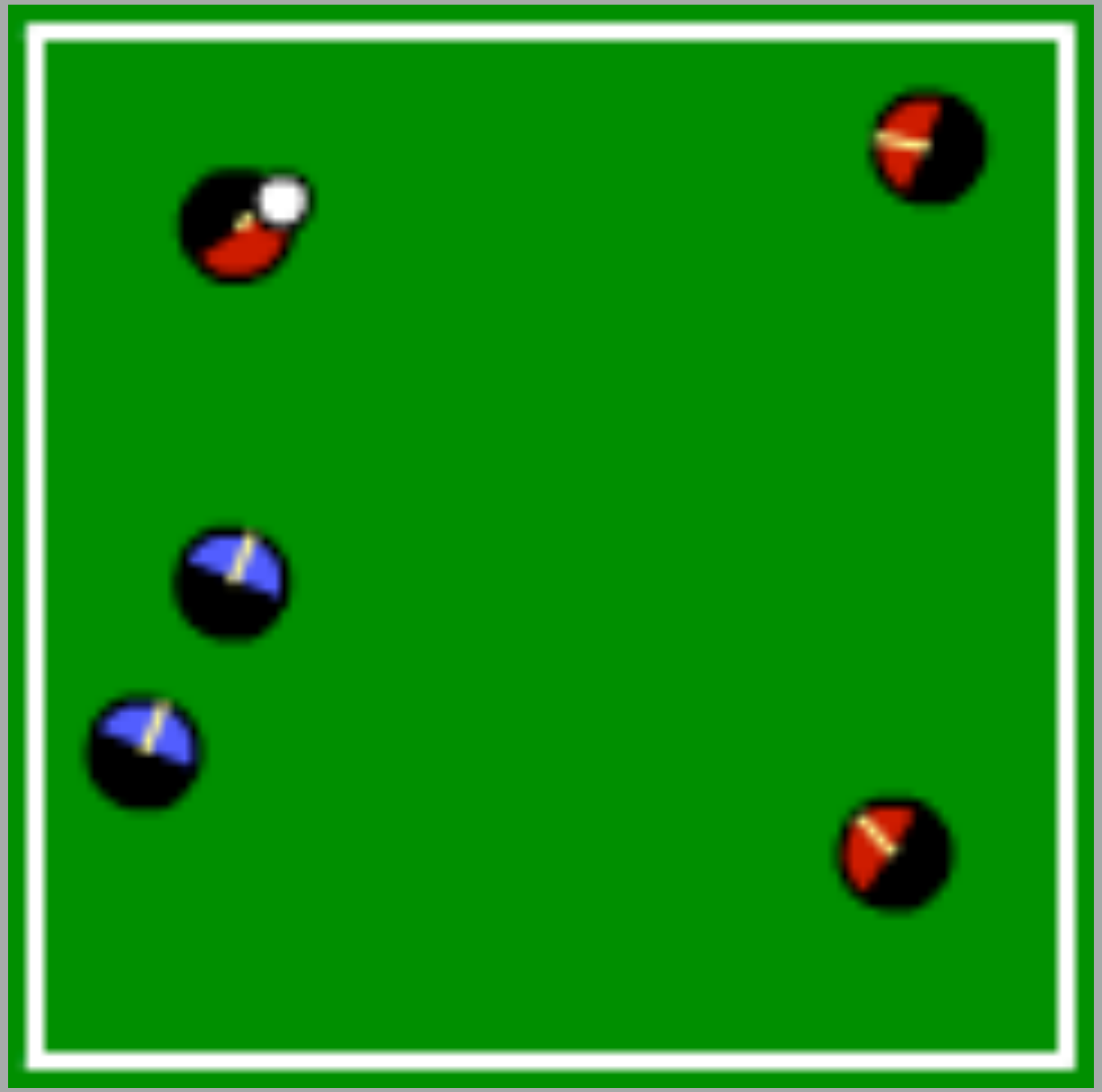


Hand-coded

- Stone & Sutton



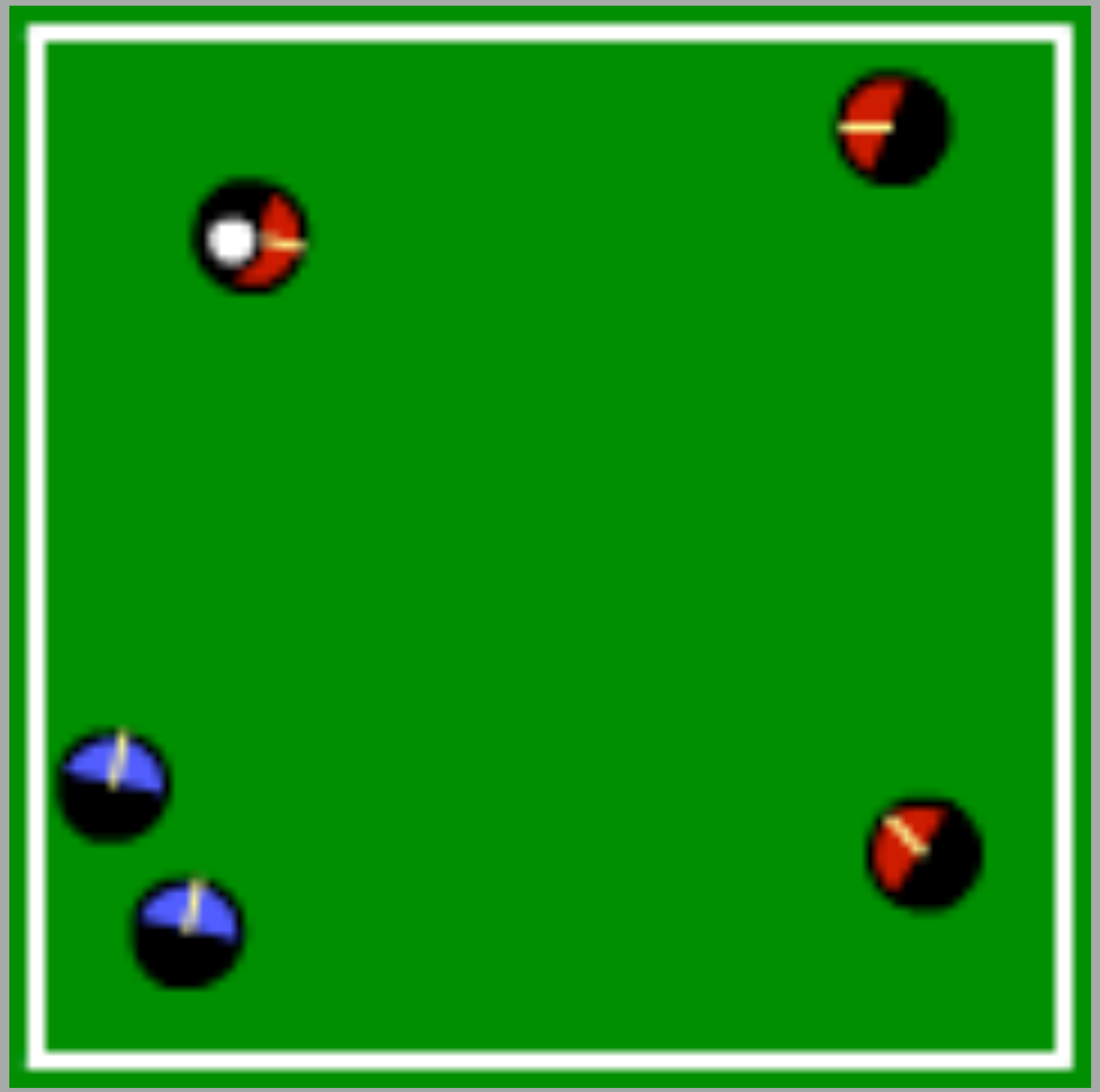
Random



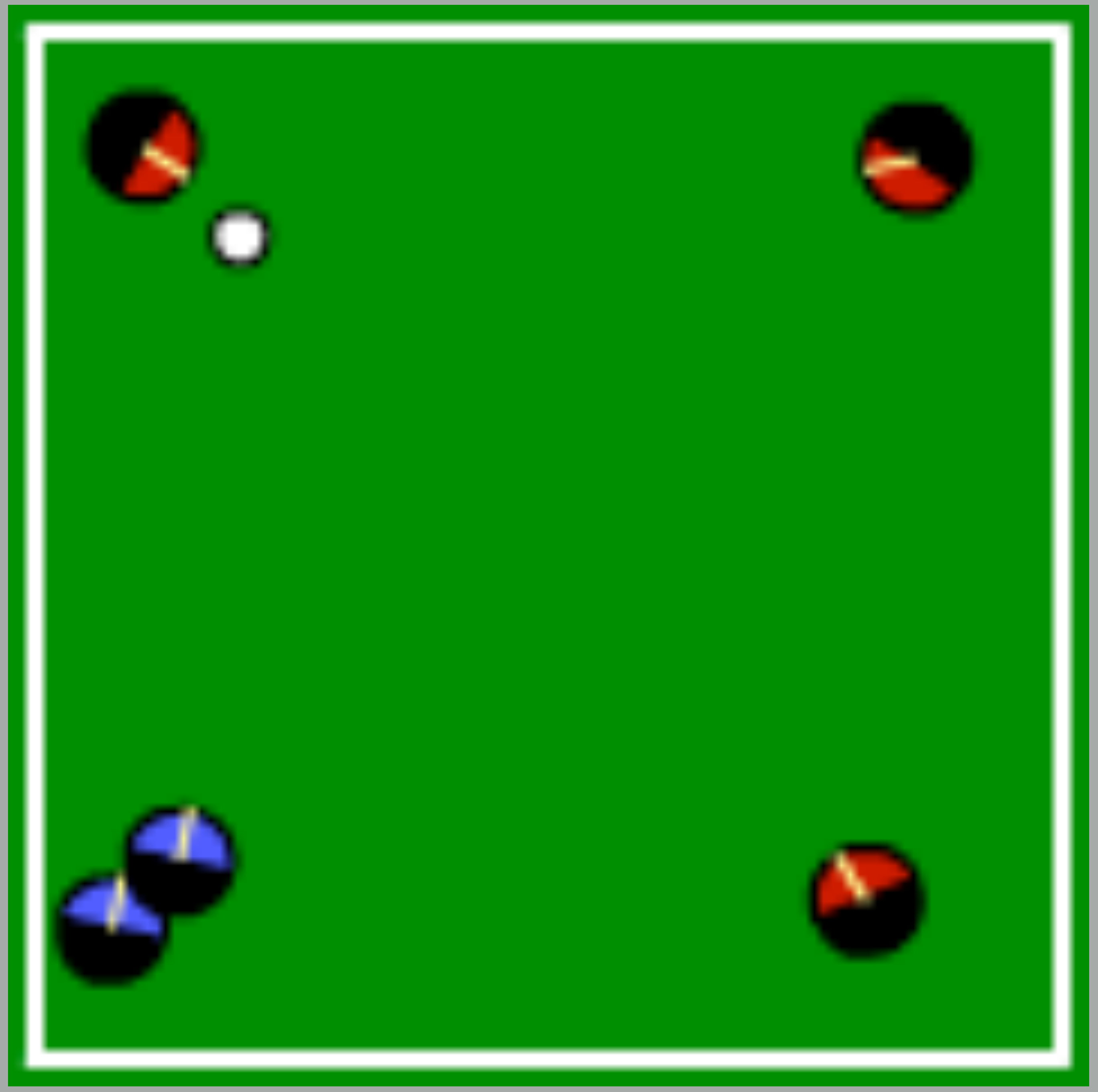
Hand-coded

Hold

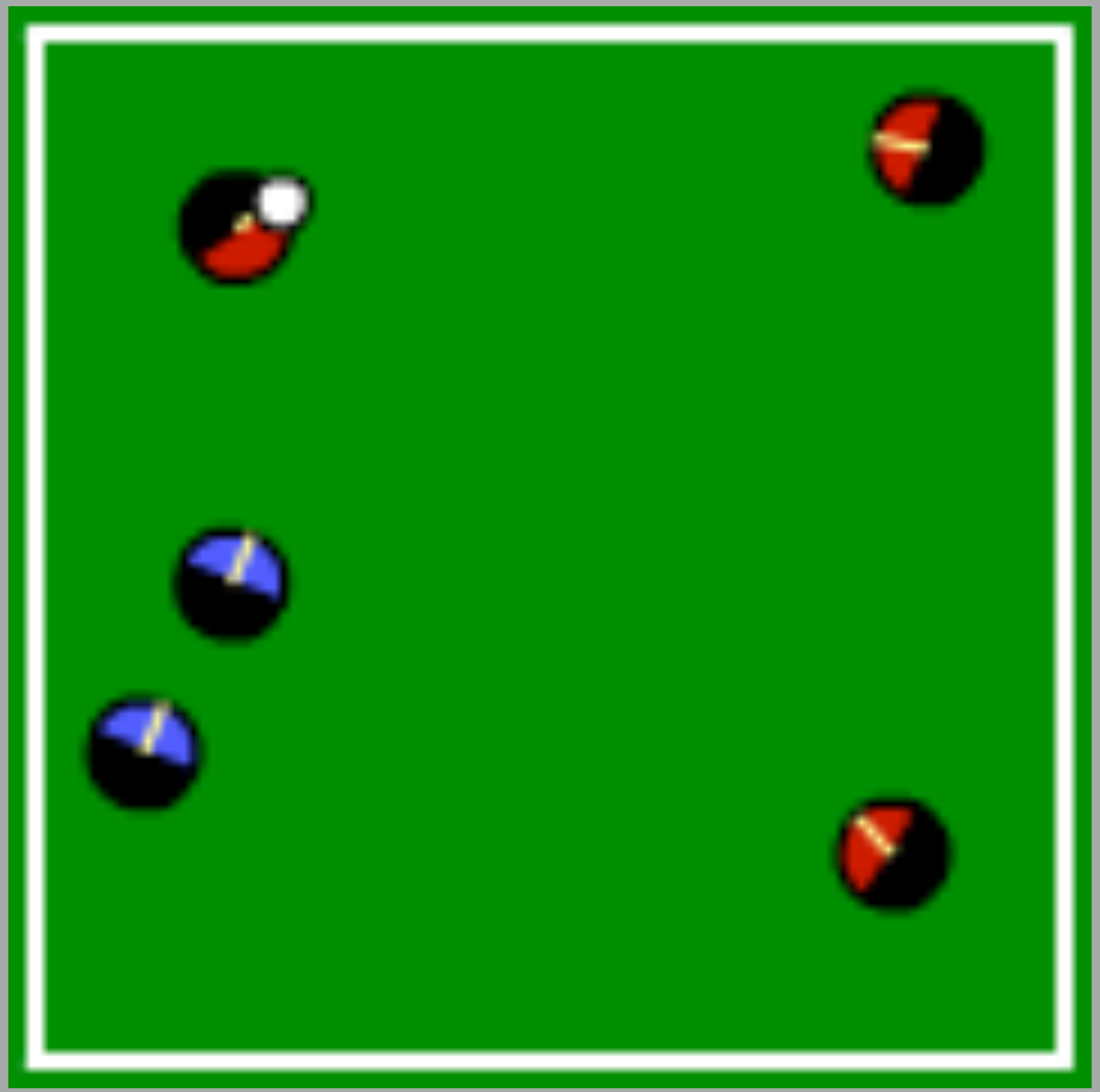
- Stone & Sutton



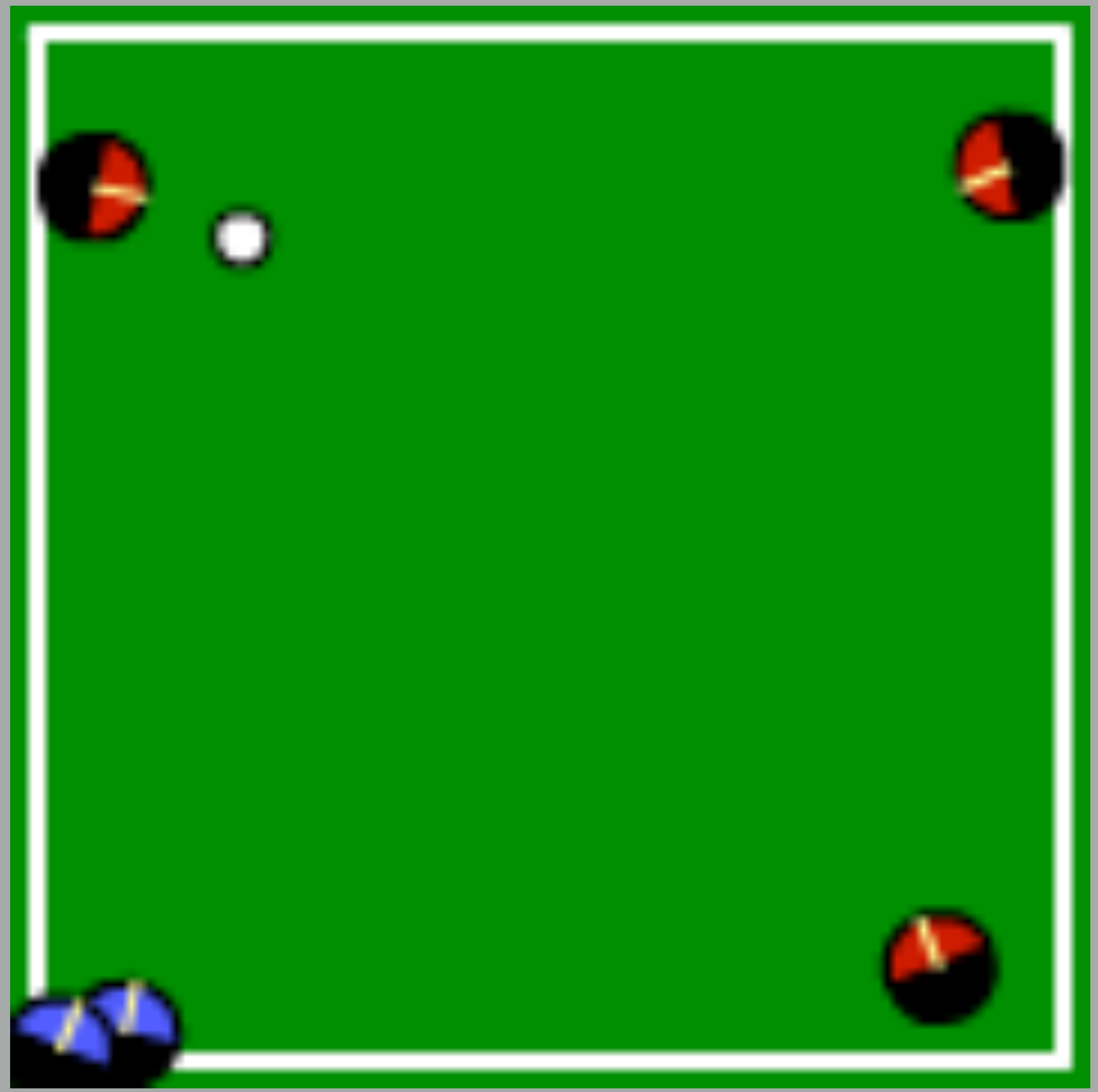
Random



Learned

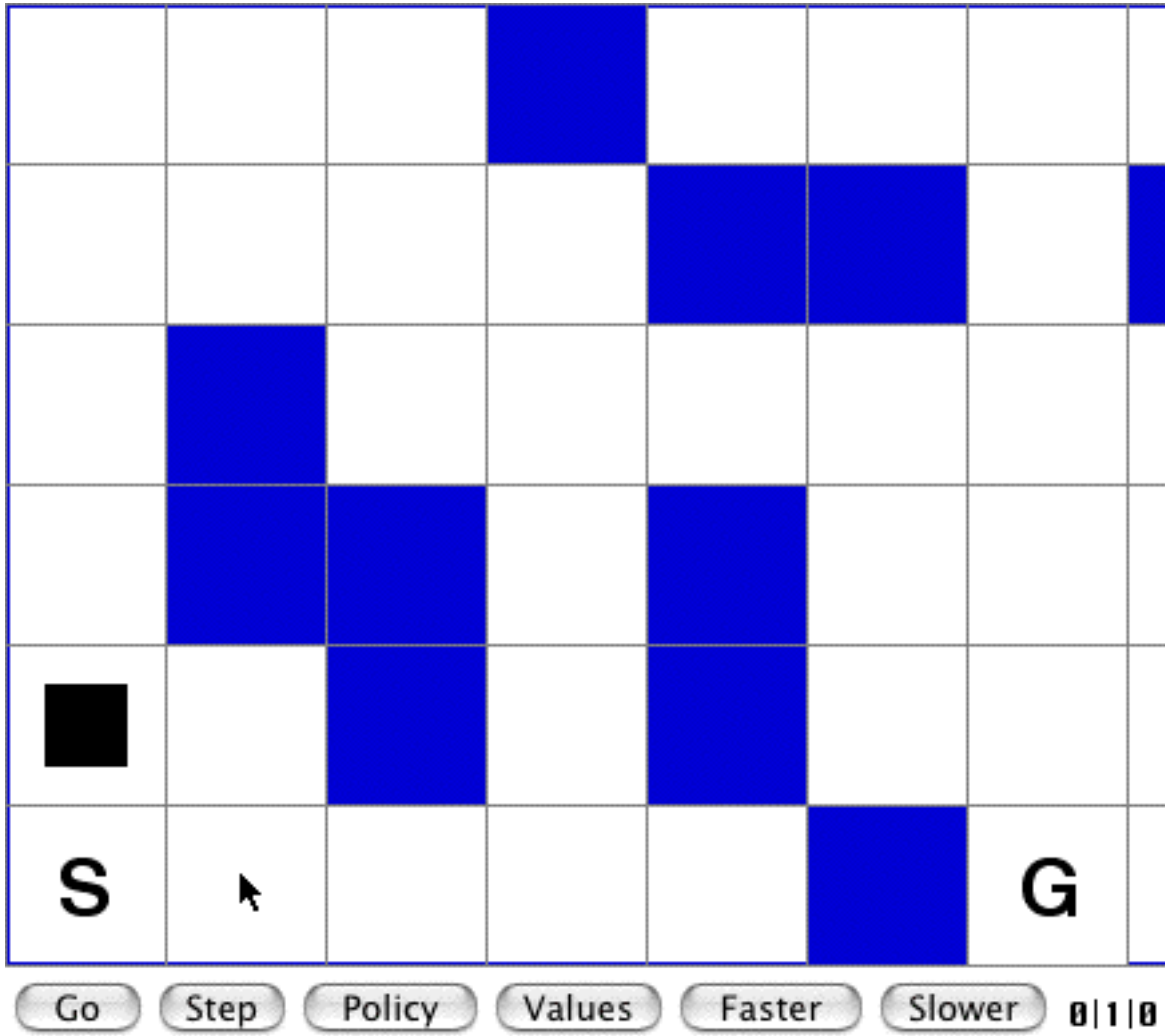


Hand-coded



Hold

GridWorld Example



Discussion advice

- Don’t ask questions about the basics of the material e.g.
  - “I didn’t understand figure 3.x”
  - “Why is there a max in the bellman optimality equation?”
- This is something you should just ask in:
  - #ta-lounge, office hours, or in class
- Ask questions you would to start a discussion with a group of people
  - e.g., is the MDP formulation limited? What problems might **not** be well framed this way?

Discussion advice

- The goal of chapter 3 is to become comfortable with the formalizations and notations of RL
- Use the math when needed to ask precise questions, using regular language will often not be precise enough
  - E.g., if you are asking about the value of a state, just write `v_{\pi}`

How many learning methods are discussed in chapter 3?

- There **exists** an optimal for an MDP
- We can **compute** the value function for a given MDP and policy
    - e.g., by hand calculations
- We can **derive** the bellman equation from the **definition** of the value function and knowledge of MDPs
- **Estimating** the value function or optimal policy typically involves an iterative algorithm or data generated from an agent interacting with an MDP
- **Approximating** the value function or optimal policy typically involves data and a function approximator

Fancy MDPs

- What about infinite MDPs?
  - Example?
  - Do we care about them?
  - What about continuous time?

Optimal Policies

- Are all optimal policies deterministic?
  - In MDPs we assume things are stationary (ie the reward function and transitions don’t change with time)
  - There can be many optimal policies. How?
  - What changes if we consider 2-player zero sum games?
    - Do we care?

State

- The meaning of Markov State
    - The agent would never do better by keeping a history of prior states and actions
- Remember MDPs are an abstract mathematical concept
  - Its a set of assumptions which may not be true in practice
  - What is important is if MDPs can be used to prove useful results and develop algs
    - e.g., Temporal difference learning

General

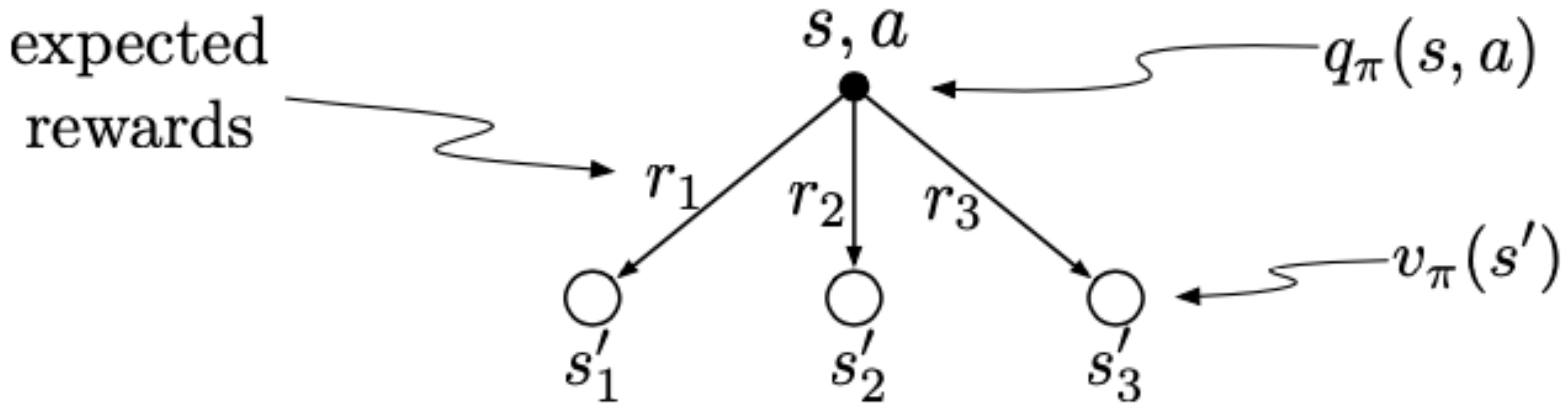
- Can we use A* or other search algorithms?
- Should such an agent still care about convergence if the environment is much bigger than the agent?
  - What do we mean here?
- How do we handle exploration in the optimal policy (Ch3 generally)?
- What is the difference between v_{\pi}(S) and q_{\pi}(S,A)?
  - Why would it be useful to ask q_{\pi} about A’s that \pi does not select?

Specifying the problem

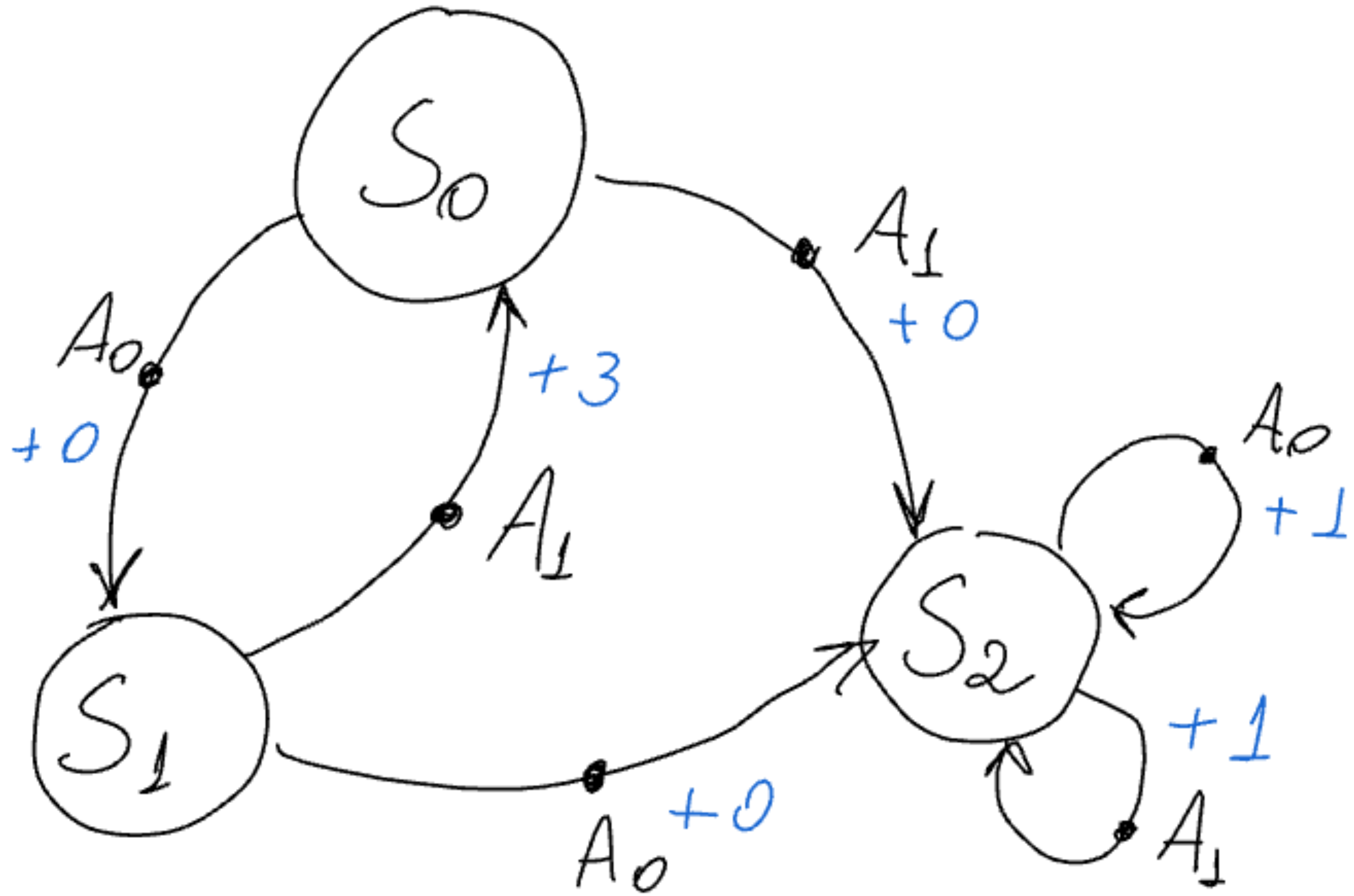
- Do we need to search for the best value of the **discount rate** also when finding the optimal policy?
  - Changing the problem is generally a bad idea
- This chapter is about problem formulations not solution methods
- Imagine learning about sorting list:
  - We know the best case performance depends on the length of the list
  - Would you ask: should we change the length of the list to get better performance?
    - NO!

Definitions vs other things

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]. \quad (3.13)$$



Solve for V*



Worksheet Review

3. (Exercise 2.2 from S&B 2nd edition) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

Worksheet Review

3. (Exercise 2.2 from S&B 2nd edition) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

T	Q1	Q2	Q3	Q4	{A*_t}	A_t	Explore?	R_1
1								
2								
3								
4								
5								

Worksheet Review

3. (Exercise 2.2 from S&B 2nd edition) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

T	Q1	Q2	Q3	Q4	{A*_t}	A_t	Explore?	R_1
1	0	0	0	0	{1,2,3,4}	1		
2								
3								
4								
5								

Worksheet Review

3. (Exercise 2.2 from S&B 2nd edition) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

T	Q1	Q2	Q3	Q4	{A*_t}	A_t	Explore?	R_1
1	0	0	0	0	{1,2,3,4}	1	Maybe	-1
2	-1	0	0	0	{2,3,4}	2	Maybe	1
3	-1	1	0	0		2		-2
4	-1	-0.5	0	0		2		2
5	-1	0.3333	0	0		3		0

Key learnings

T	Q1	Q2	Q3	Q4	{A*_t}	A_t	Explore?	R_1
1	0	0	0	0	{1,2,3,4}	1	Maybe	-1
2	-1	0	0	0	{2,3,4}	2	Maybe	1
3	-1	1	0	0		2		-2
4	-1	-0.5	0	0		2		2
5	-1	0.3333	0	0		3		0

- Initial Q-values (all zeros) do not impact the computation of the sample average
- We don't change the values of actions not taken
- The explore step or \epsilon step might choose the greedy action: can only know for sure when agent explores
- Try to imagine the agents life:
  - Look at the Q-values; pick an action; observe the reward; update one of the Q-values
  - Look at the Q-values; pick an action; observe the reward; update one of the Q-values
  - …. forever

Worksheet Review

1. Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2$, $R_2 = -2$, $R_3 = 0$ followed by an infinite sequence of 7s. What are G_1 and G_0?

Worksheet Review

1. Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2, R_2 = -2, R_3 = 0$ followed by an infinite sequence of 7s. What are G_1 and G_0?

- Work backwards
- Sequence of rewards is: R1 = 2, R2 = -2, R3 = 0, R4 = 7, R5 = 7, …..
- Let's start with R4=7, the first of the unending sequence of 7's
- Using our formula G_t = R_{t+1} + gamma R_{t+2} + gamma^2 R_{t+3} + …
  we write it as:
  - G_3 = R_4 + gamma R_5 + gamma^2 R_6 + gamma^3 R_7 + …
    = 7 + 0.9*7 + (0.9)^2*7 + + (0.9)^3*7…
  - Use our special formula to work out G_3: $\sum_{k=0}^{\infty} \gamma^k R = \frac{R}{1 - \gamma}$
-

Worksheet Review

1. Suppose $\gamma = 0.9$ and the reward sequence is $R_1 = 2, R_2 = -2, R_3 = 0$ followed by an infinite sequence of 7s. What are G_1 and G_0?

- Continue working backwards from G_3 using our other special formula
  - $G_t = R_{t+1} + \gamma G_{t+1}$

Answer:

$$G_3 = 7 + 0.9 \times 7 + 0.9^2 \times 7 + ... = 7 \times \frac{1}{1-0.9} = 70$$
$$G_2 = R_3 + 0.9 \times G_3 = 0 + 0.9 \times 70 = 63$$
$$G_1 = R_2 + 0.9 \times G_2 = -2 + 0.9 \times 63 = 54.7$$
$$G_0 = R_1 + 0.9 \times G_1 = 2 + 0.9 \times 54.7 = 51.23$$

Worksheet Question

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left| \sum_{i=0}^{\infty} \gamma^i R_{t+1+i} \right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

Recall $\displaystyle\sum_{i=0}^{\infty} |a_i| < \infty$ then $\displaystyle\left| \sum_{i=0}^{\infty} a_i \right| < \infty$

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty \qquad\qquad \text{for } \gamma \in [0, 1)$$

If $\displaystyle\sum_{i=0}^{\infty} |\gamma^i R_{t+1+i}| < \infty$ then $\displaystyle\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty$

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

If $\sum_{i=0}^{\infty} |\gamma^i R_{t+1+i}| < \infty$ then $\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty$

$$= \sum_{i=0}^{\infty} \gamma^i |R_{t+1+i}|$$

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left| \sum_{i=0}^{\infty} \gamma^i R_{t+1+i} \right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

If $\sum_{i=0}^{\infty} |\gamma^i R_{t+1+i}| < \infty$ then $\left| \sum_{i=0}^{\infty} \gamma^i R_{t+1+i} \right| < \infty$

$$= \sum_{i=0}^{\infty} \gamma^i |R_{t+1+i}|$$

$$\leq \sum_{i=0}^{\infty} \gamma^i R_{\max}$$

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

If $\sum_{i=0}^{\infty} |\gamma^i R_{t+1+i}| < \infty$ then $\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty$

$$= \sum_{i=0}^{\infty} \gamma^i |R_{t+1+i}|$$

$$\leq \sum_{i=0}^{\infty} \gamma^i R_{\max}$$

$$= R_{\max} \sum_{i=0}^{\infty} \gamma^i$$

4. Prove that the discounted sum of rewards is always finite, if the rewards are bounded: $|R_{t+1}| \leq R_{\max}$ for all t for some finite $R_{\max} > 0$.

$$\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty \qquad \text{for } \gamma \in [0, 1)$$

If $\displaystyle\sum_{i=0}^{\infty} |\gamma^i R_{t+1+i}| < \infty$ then $\displaystyle\left|\sum_{i=0}^{\infty} \gamma^i R_{t+1+i}\right| < \infty$

$$= \sum_{i=0}^{\infty} \gamma^i |R_{t+1+i}|$$

$$\leq \sum_{i=0}^{\infty} \gamma^i R_{\max}$$

$$= R_{\max} \sum_{i=0}^{\infty} \gamma^i = \frac{R_{\max}}{1 - \gamma}$$

$R_{\max}$ and $\frac{1}{1-\gamma}$ are finite.