Register at: essai.si

ESSAI & ACAI 2023
LJUBLJANA, SLOVENIA

MODEL UNCERTAINTY IN SEQUENTIAL DECISION MAKING

DAVID PARKER
University of Oxford

BRUNO LACERDA
University of Oxford

NICK HAWES
University of Oxford

Recap

- Uncertain MDPs with rectangularity assumptions
  - MDPs plus epistemic uncertainty: set of transition functions
  - control policies + robust control
  - environment policies - static vs dynamic uncertainty
  - robust value iteration (robust dynamic programming)
  - implementation with interval MDPs (IMDPs)
  - non-memoryless policies (static uncertainty)
  - generating / learning intervals
  - uncertainty set representations
  - tool support: PRISM

Course contents

- ~~Markov decision processes (MDPs) and stochastic games~~
  - ~~MDPs: key concepts and algorithms~~
  - ~~stochastic games: adding adversarial aspects~~
- ~~Uncertain MDPs~~
  - ~~MDPs + epistemic uncertainty, robust control, robust dynamic programming, interval MDPs, uncertainty set representation, challenges, tools~~
- Sample-based uncertain MDPs
  - removing the transition independence assumption
- Bayes-adaptive MDPs
  - maintaining a distribution over the possible models

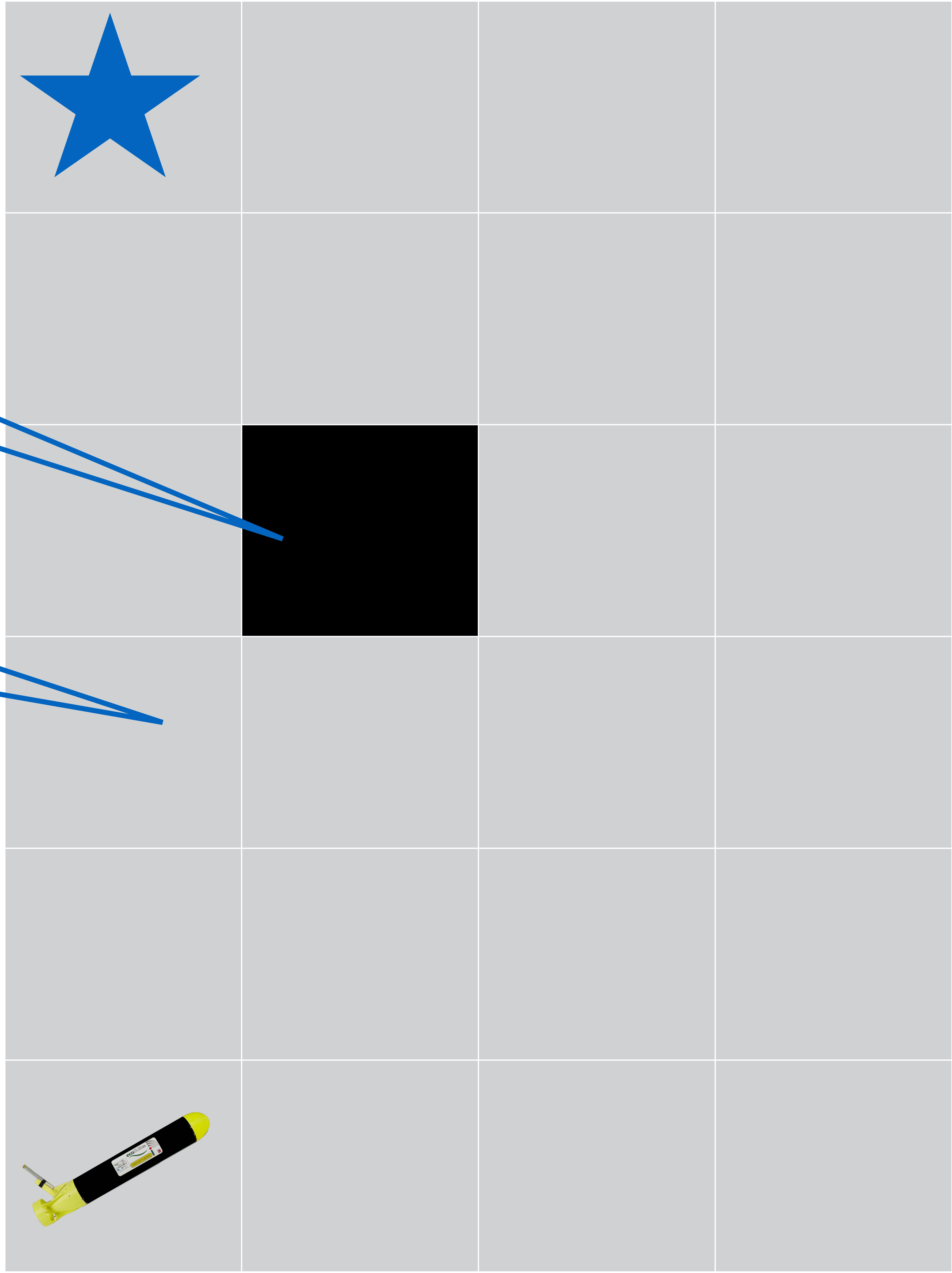
Sample-based UMDPs

SSP revisited

$$\mathcal{M} = (S, s_0, A, P, C, goal)$$

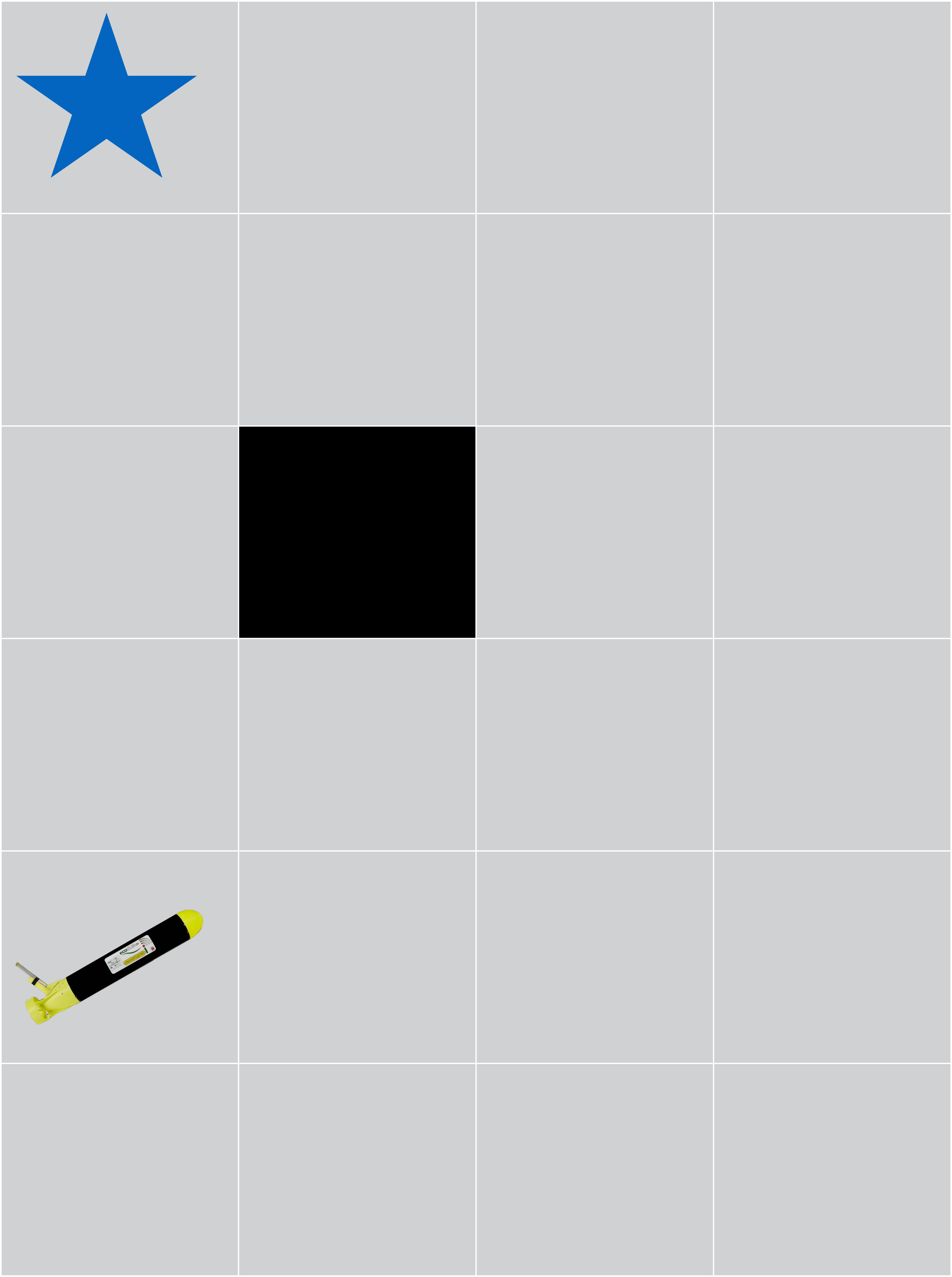
- SSP: Minimise the expected cost of reaching a target state set $goal \subseteq S$
  - for a cost function $C : S \times A \rightarrow \mathbb{R}_{\geq 0}$
  - minimise $V^\pi(s) = \mathbb{E}_s^\pi(X^C)$ where $X^C(s_0 a_0 s_1 a_1 \ldots) = \Sigma_{i=0}^\infty C(s_i, a_i)$
- Assumptions for SSP
  - $goal$ states are absorbing and zero-cost
  - there is a proper policy (i.e., which reaches $goal$ with probability 1 from all states)
  - every improper policy incurs an infinite cost from every state
    from which it does not reach $goal$ with probability 1

Example

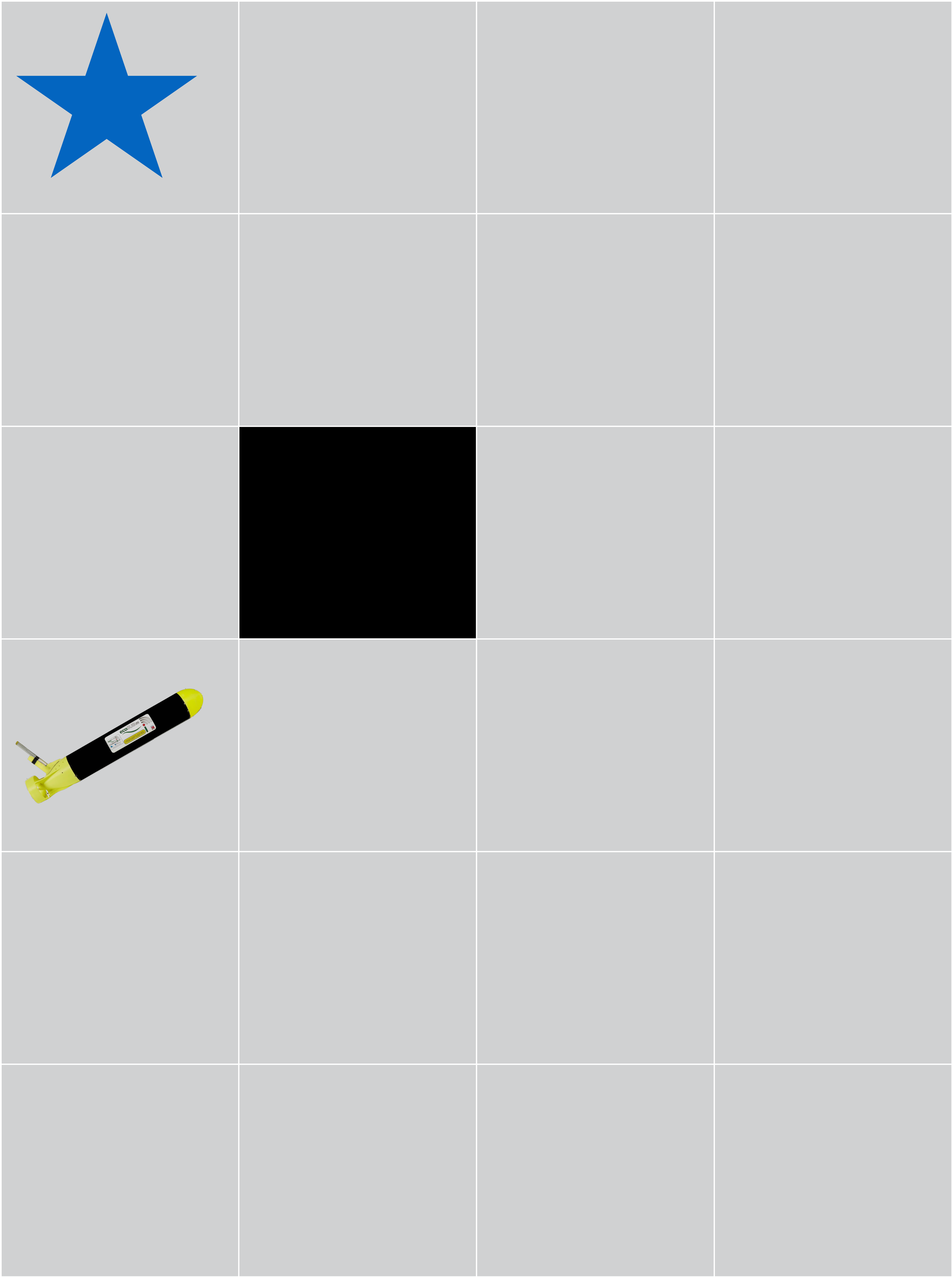


- Reach goal location subject to currents
- Mission to be executed between 6am and 6pm
  - Navigation policy must perform well under all environment conditions

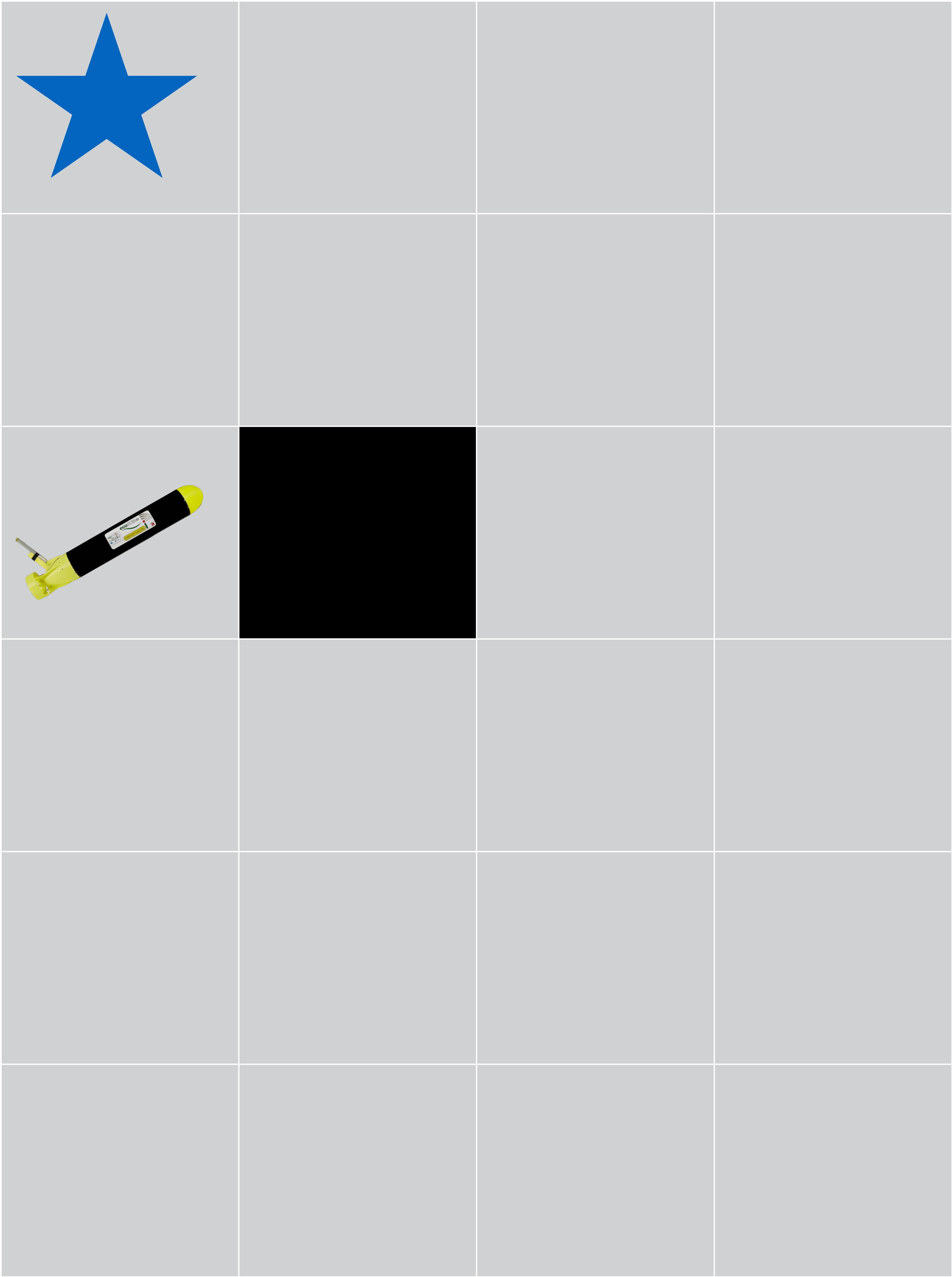
Example



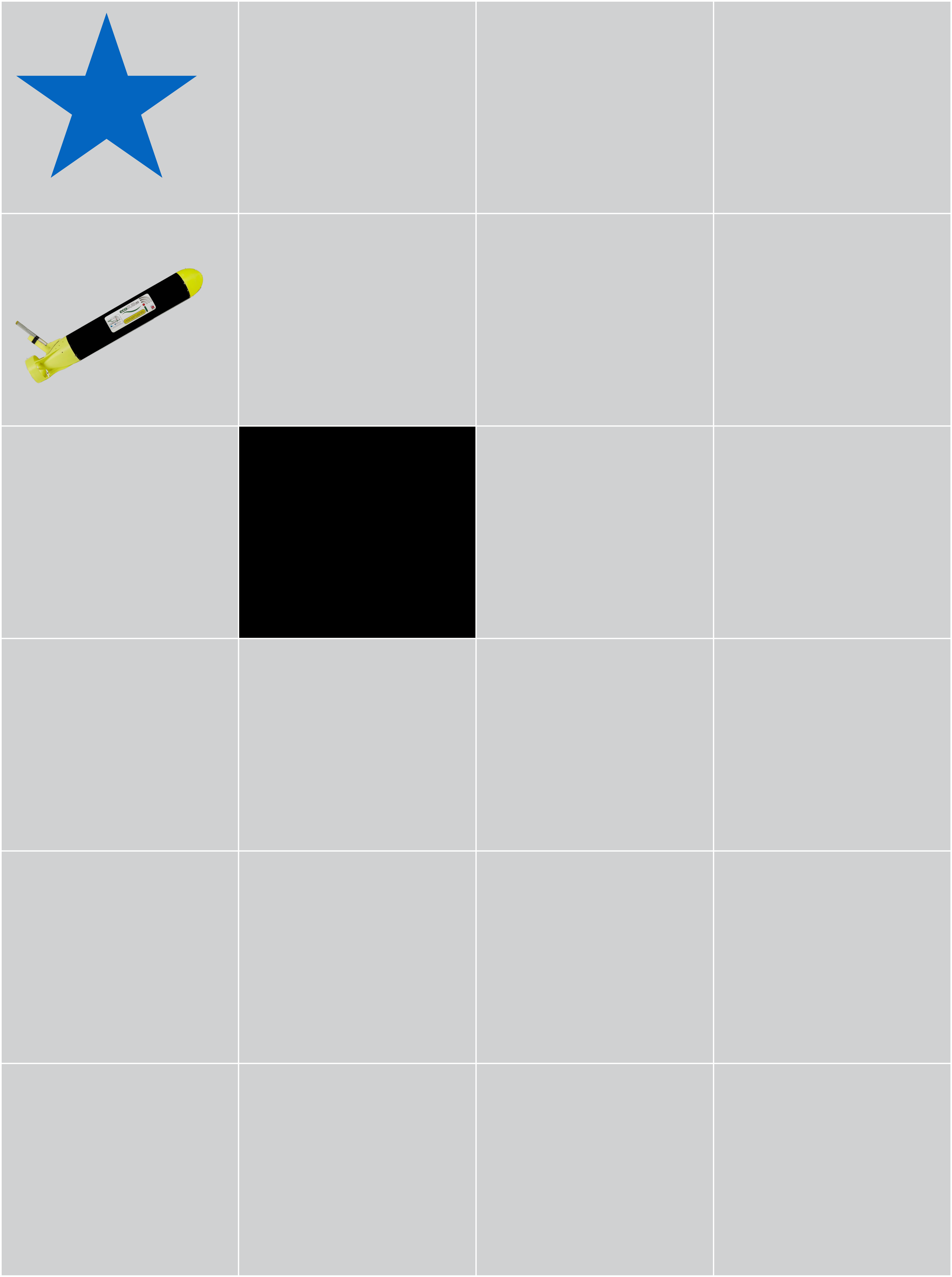
Example



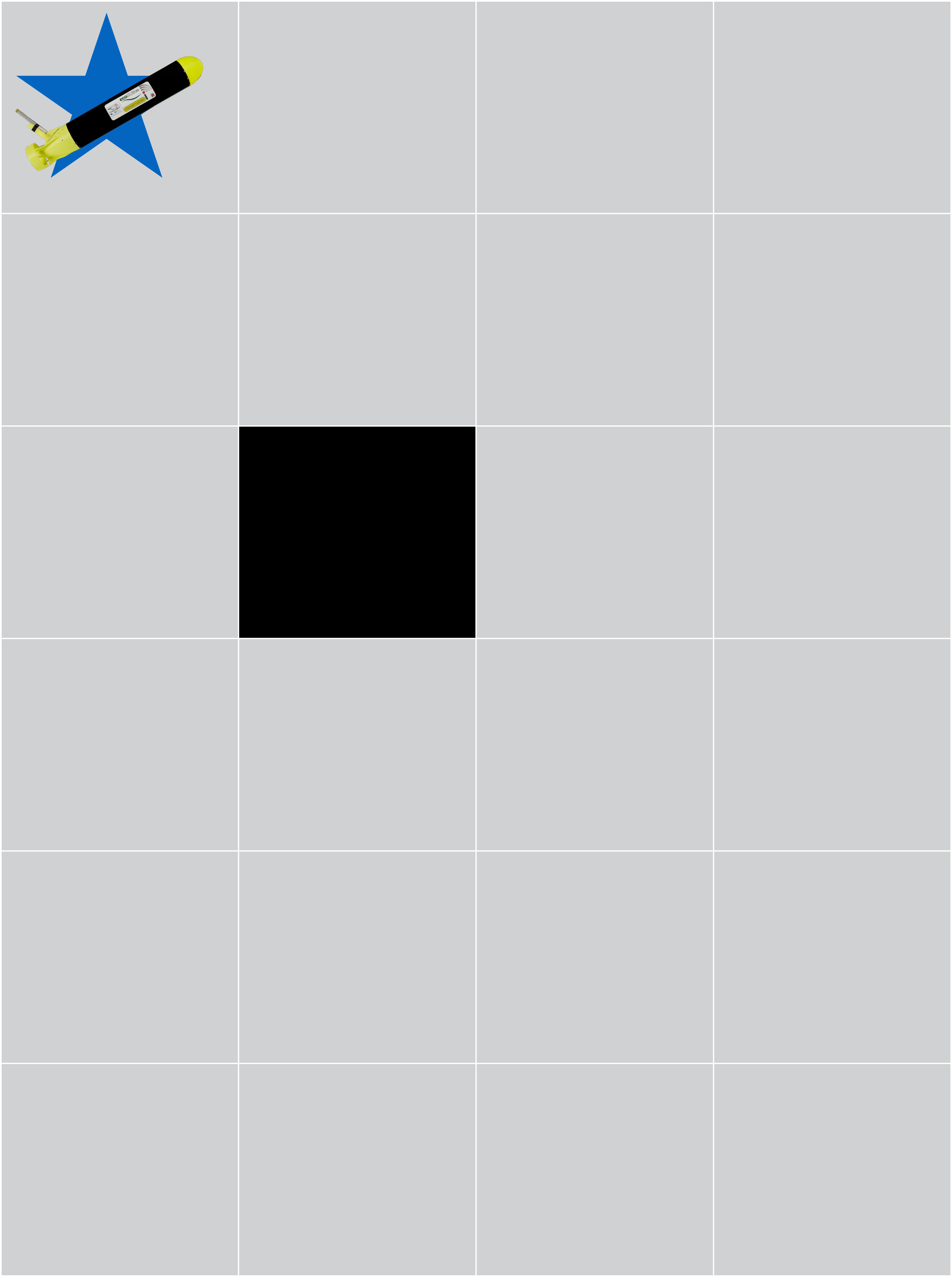
Example



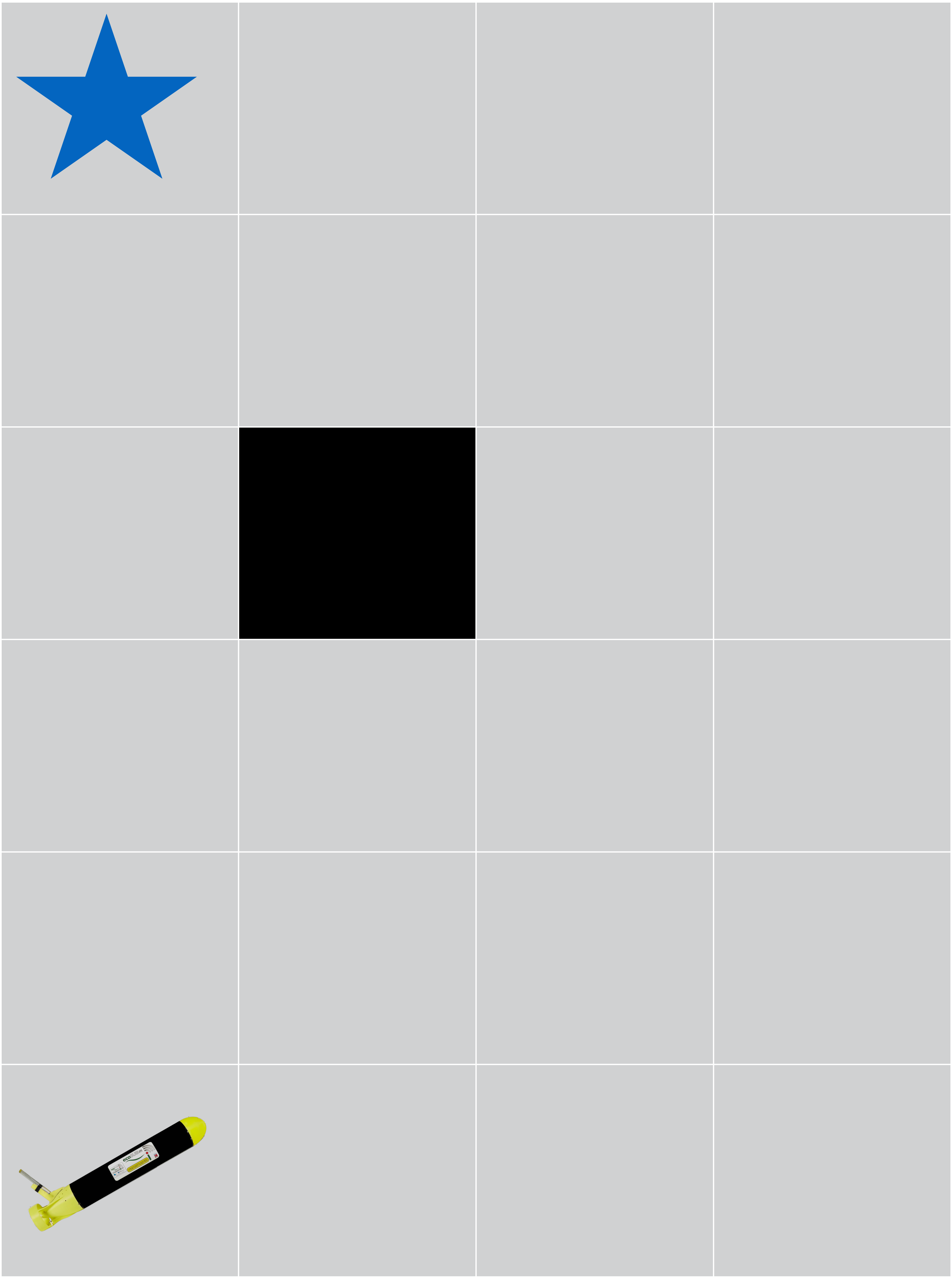
Example



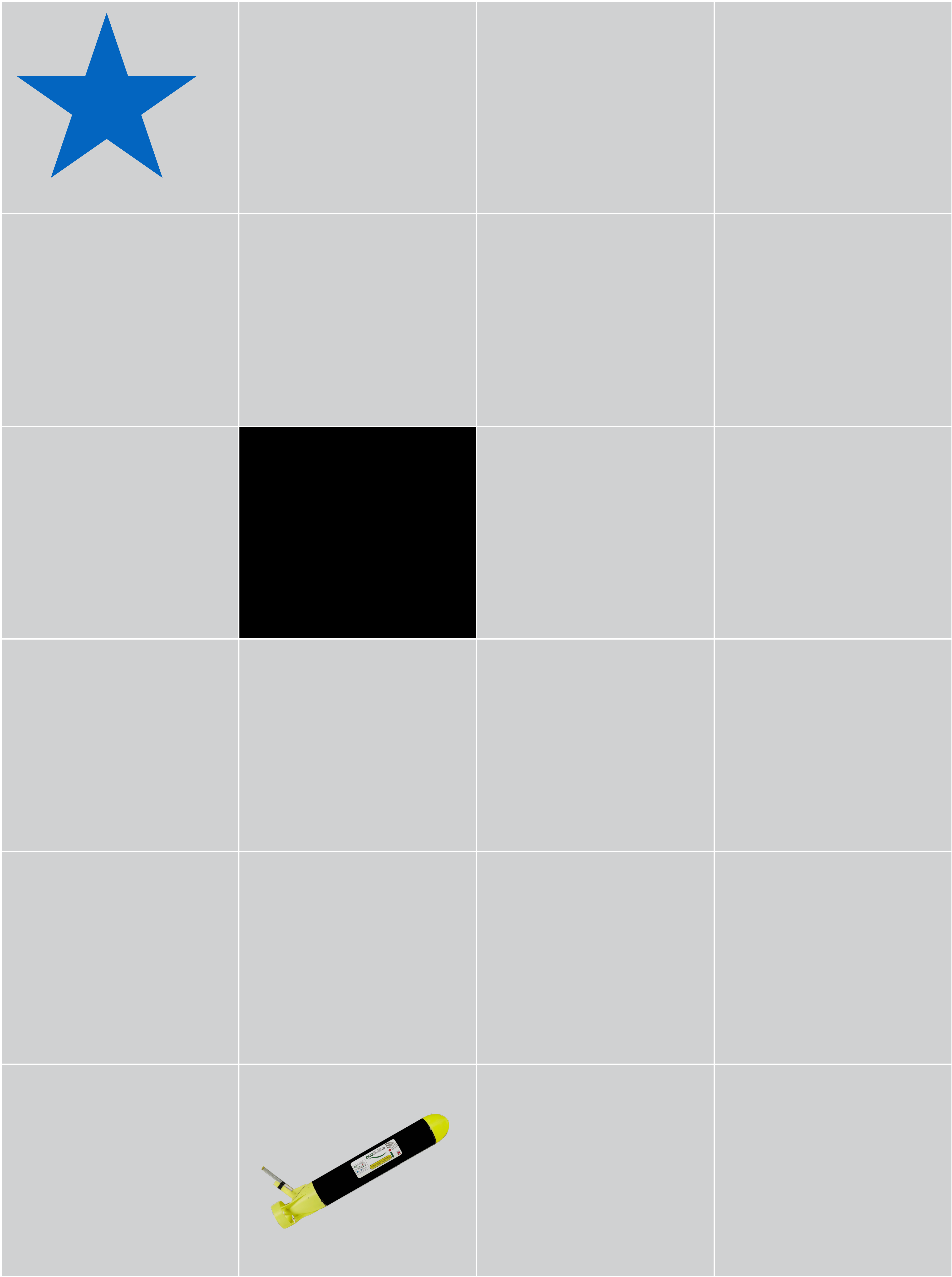
Example



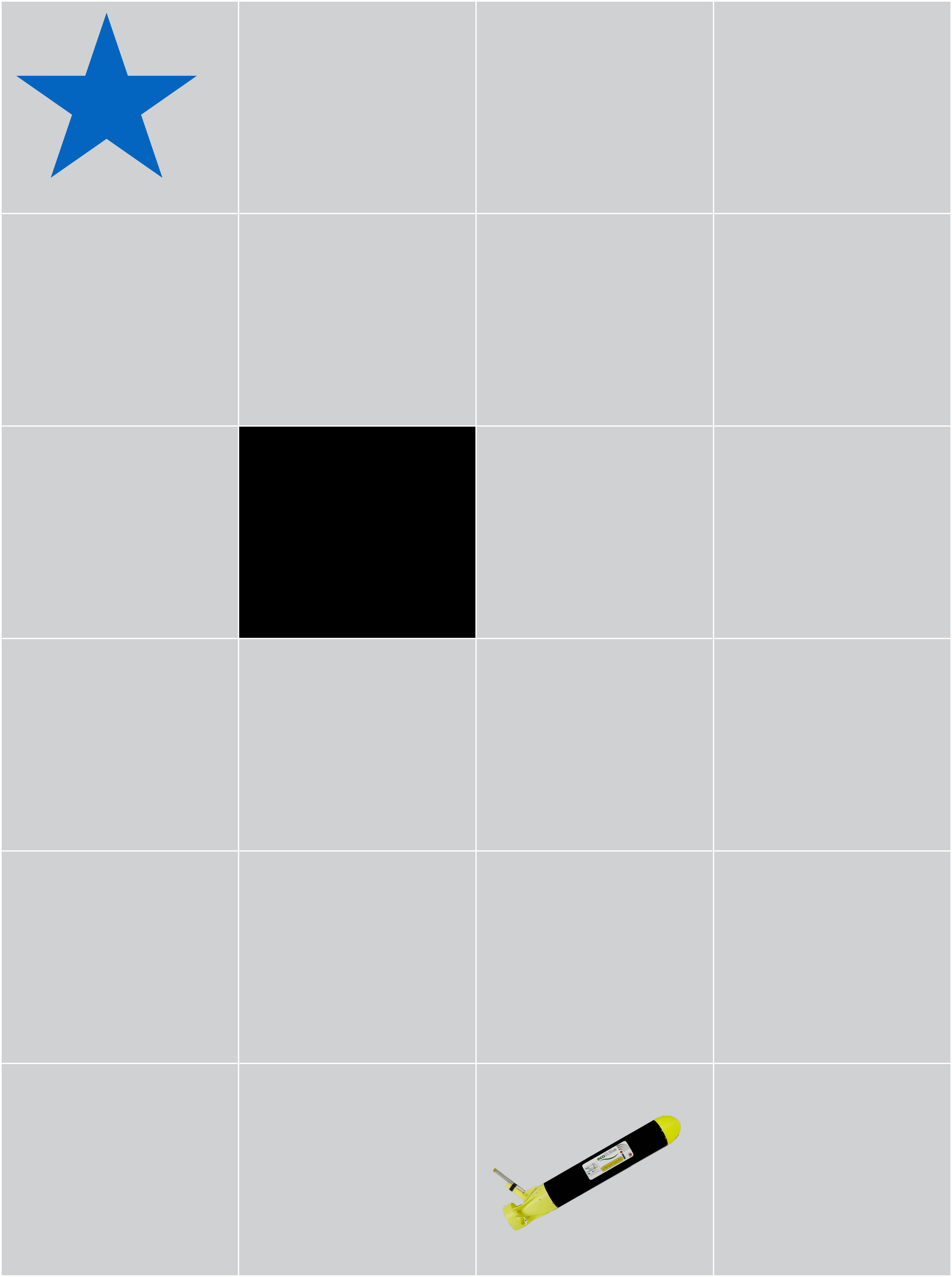
Example



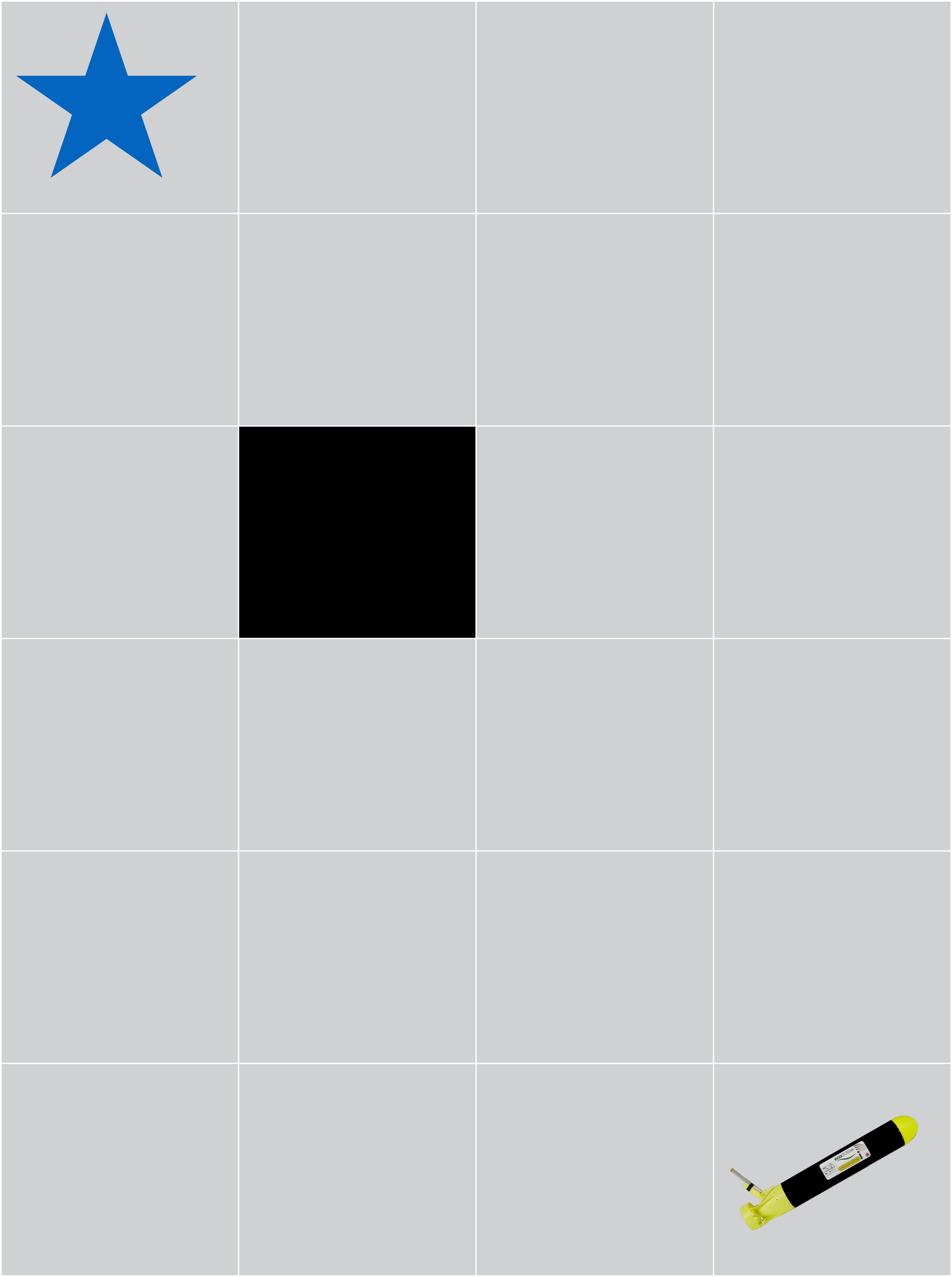
Example



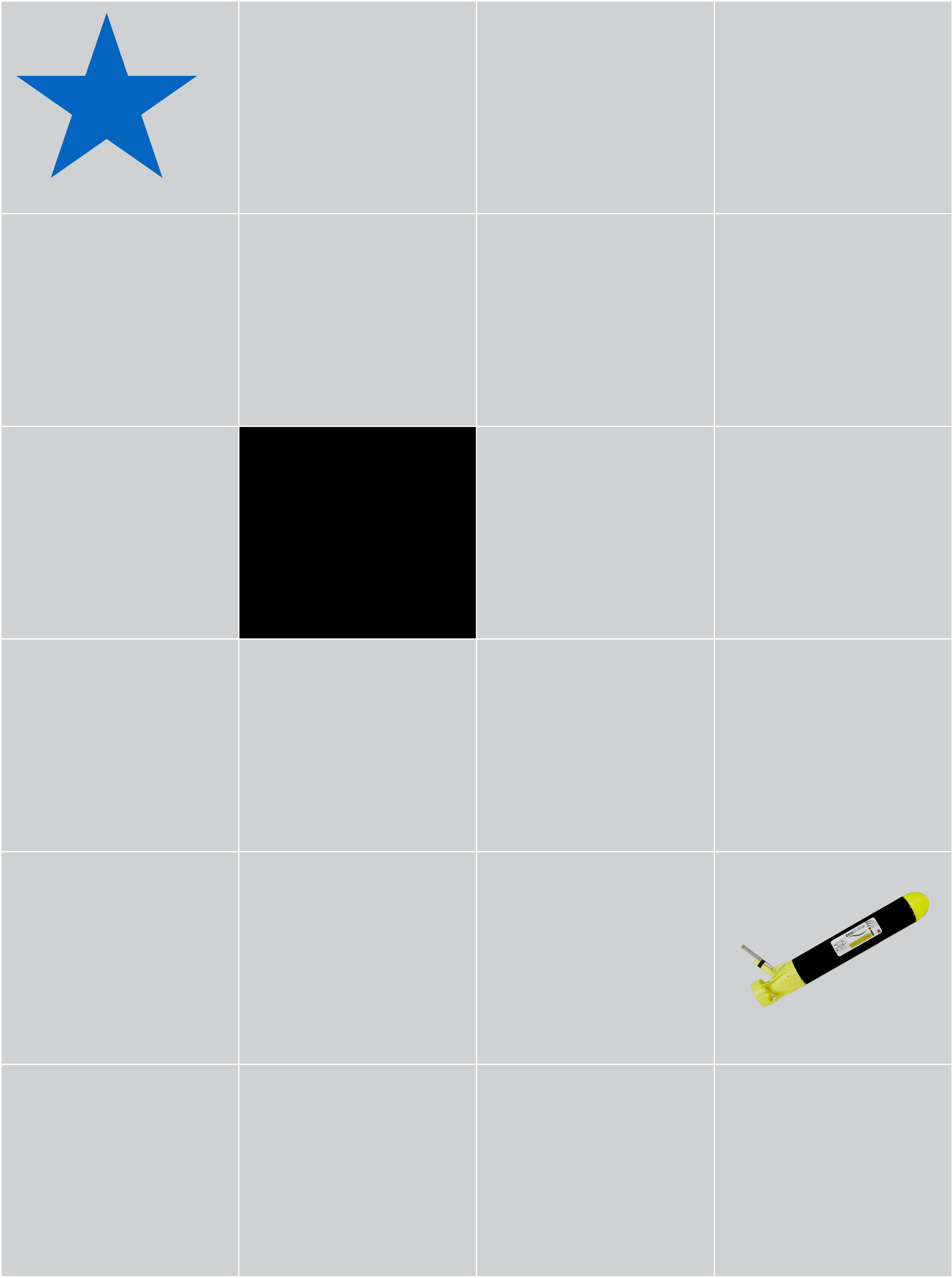
Example



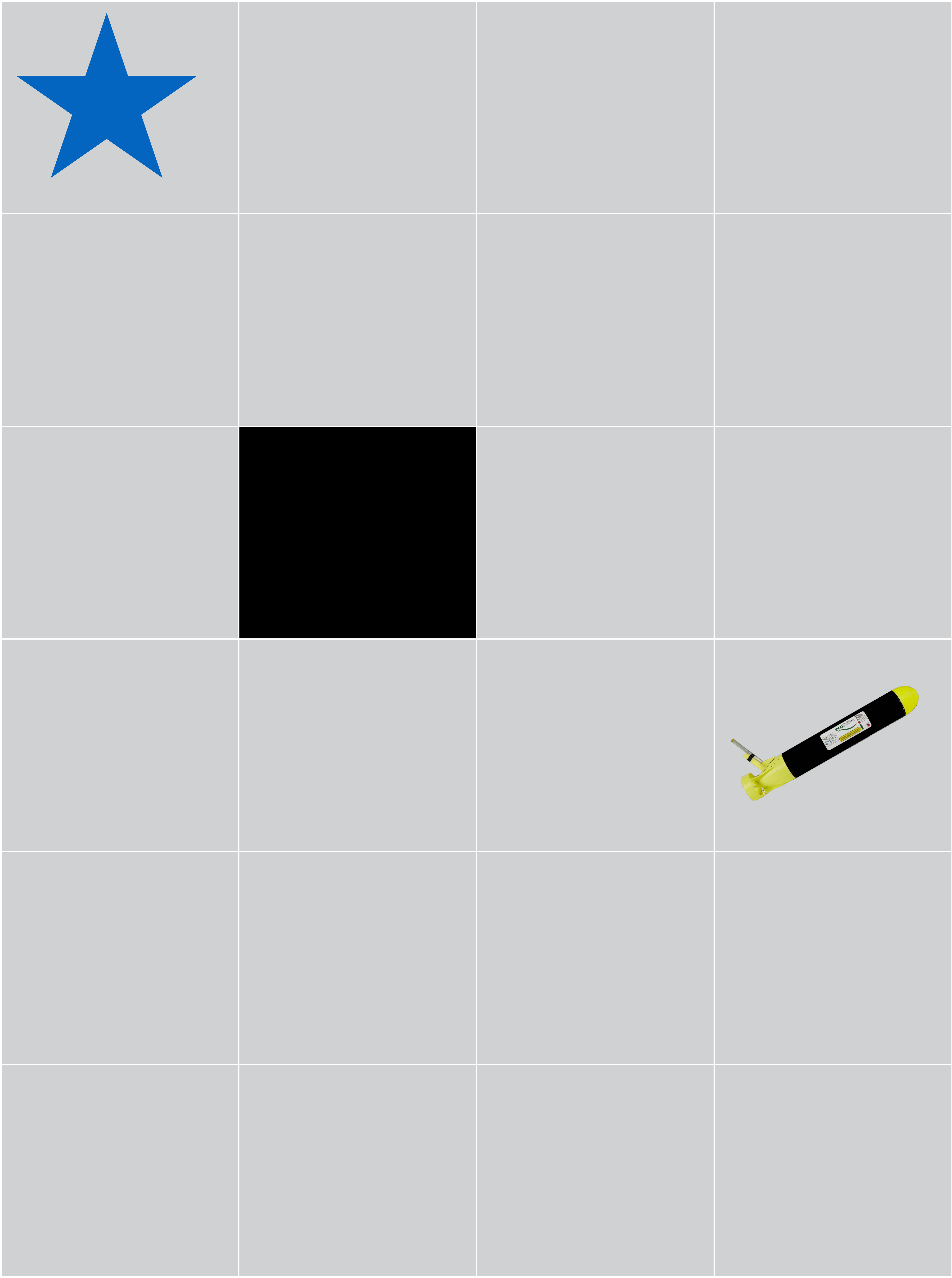
Example



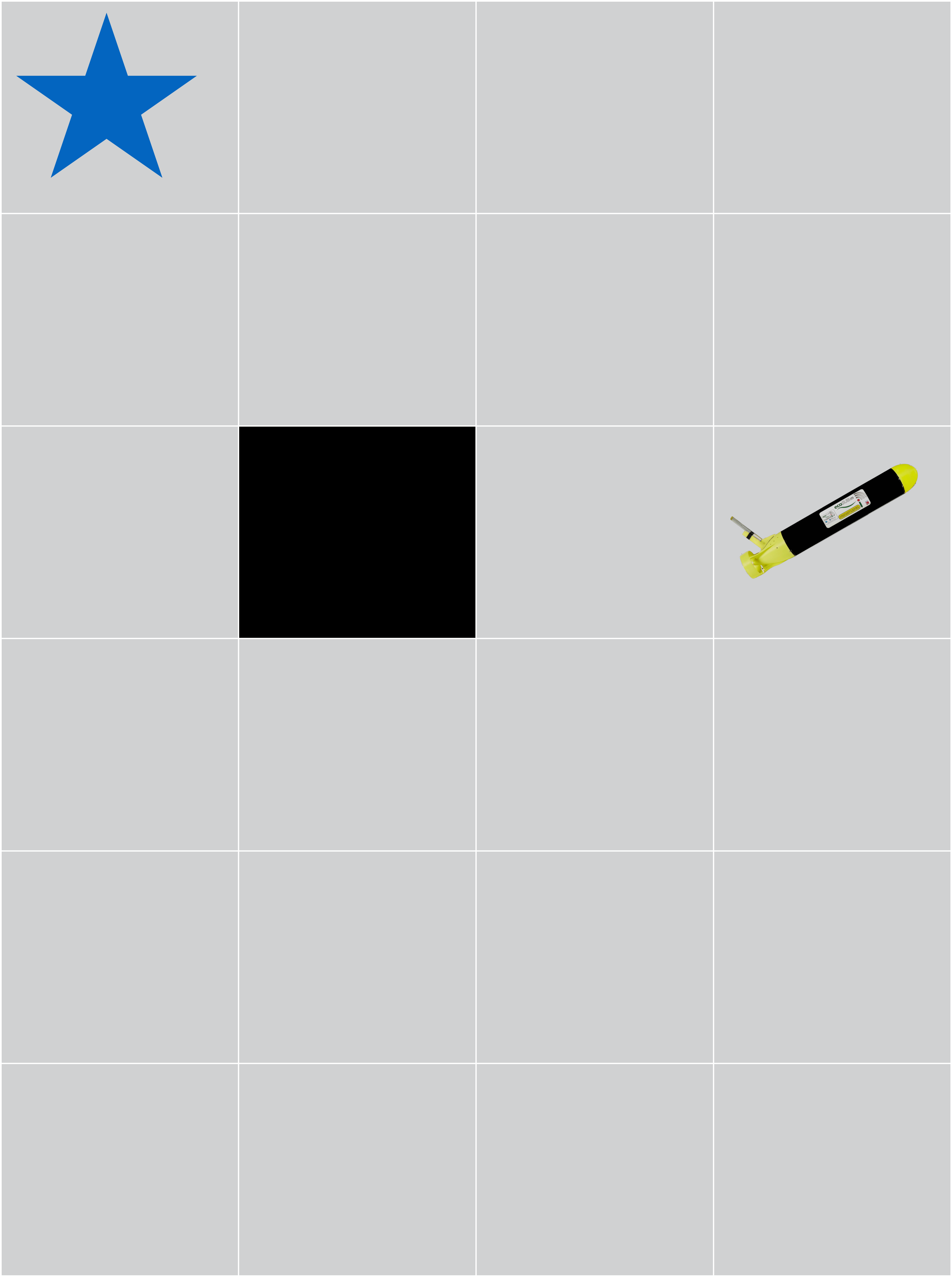
Example



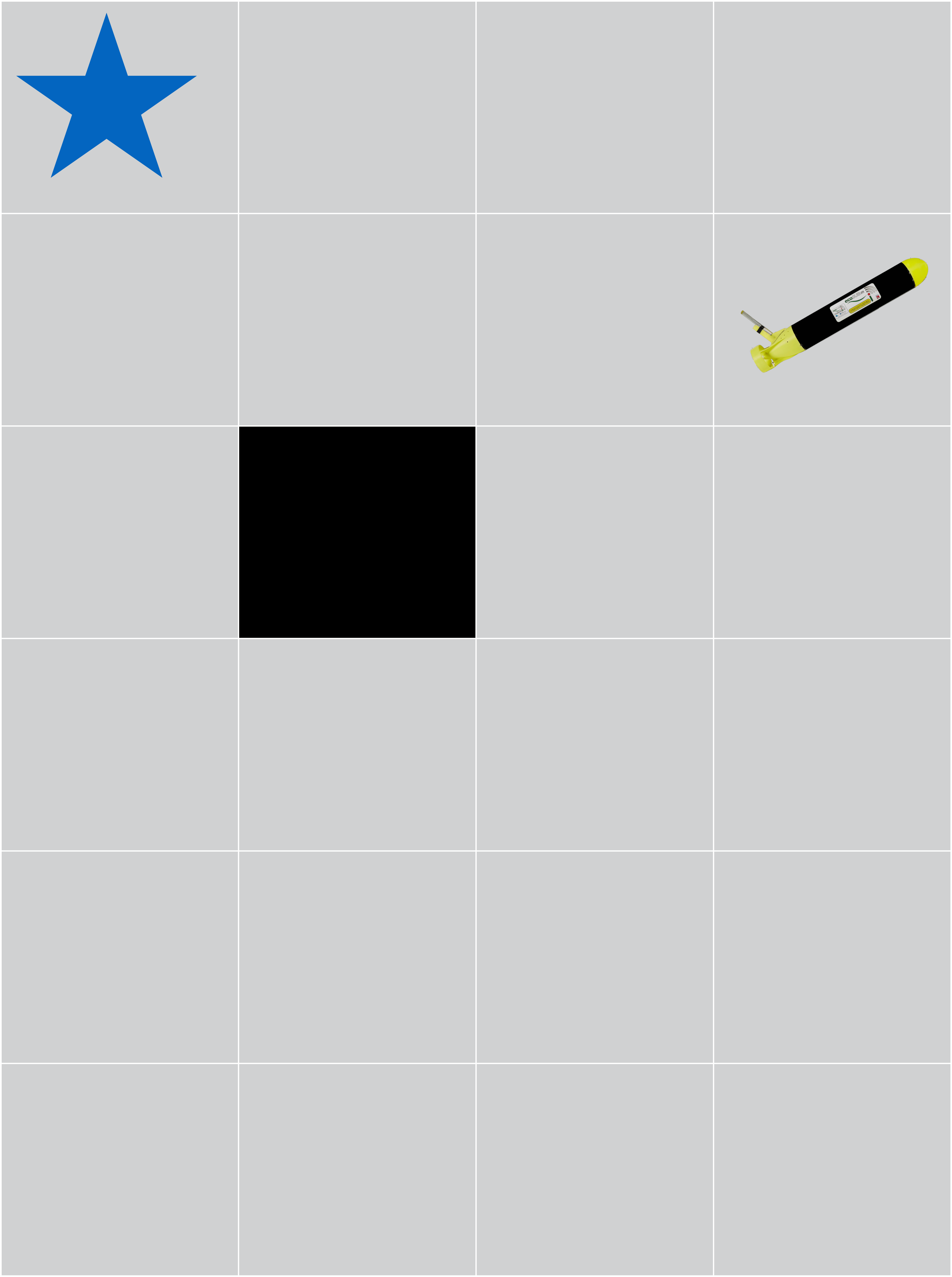
Example



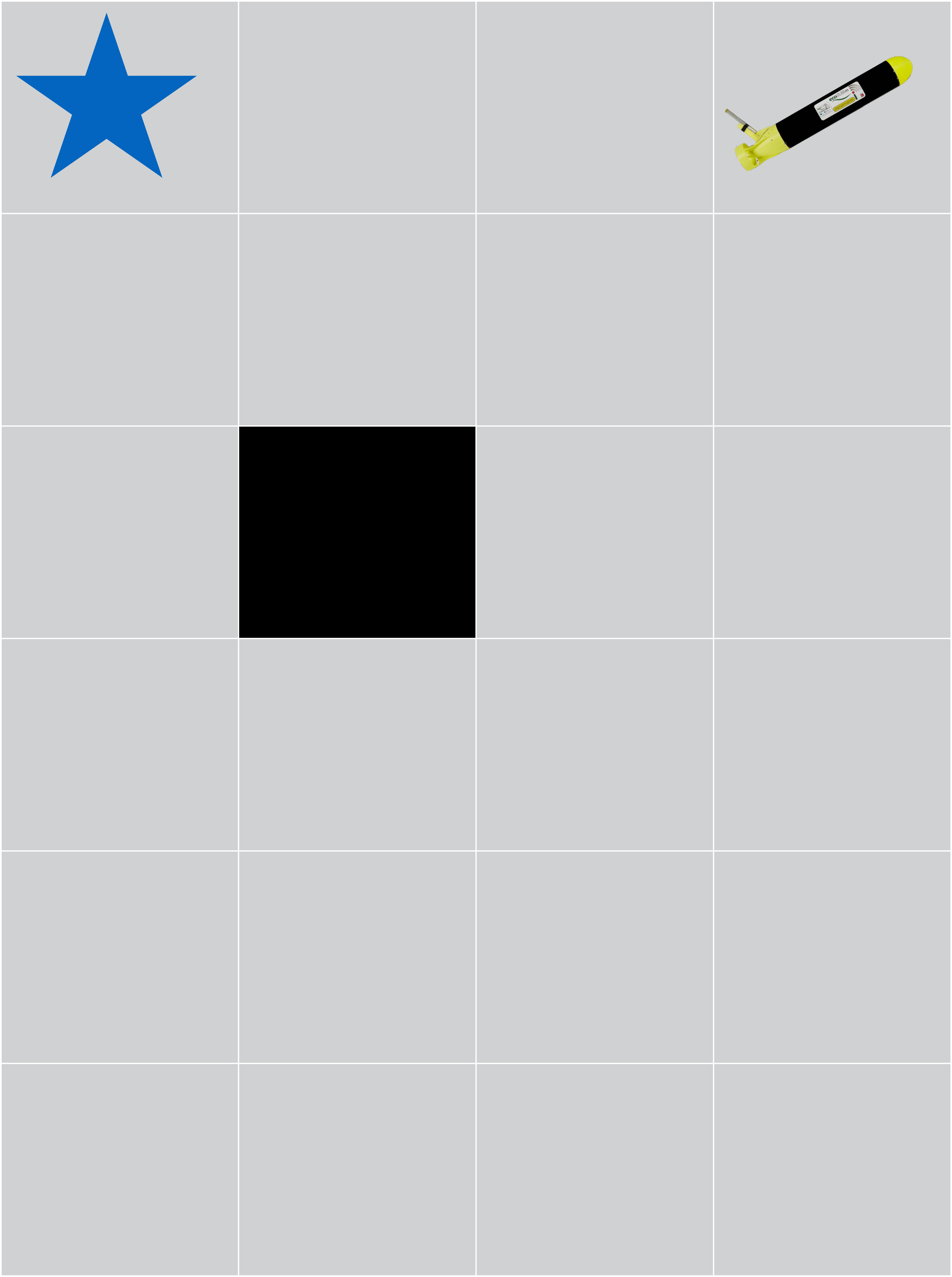
Example



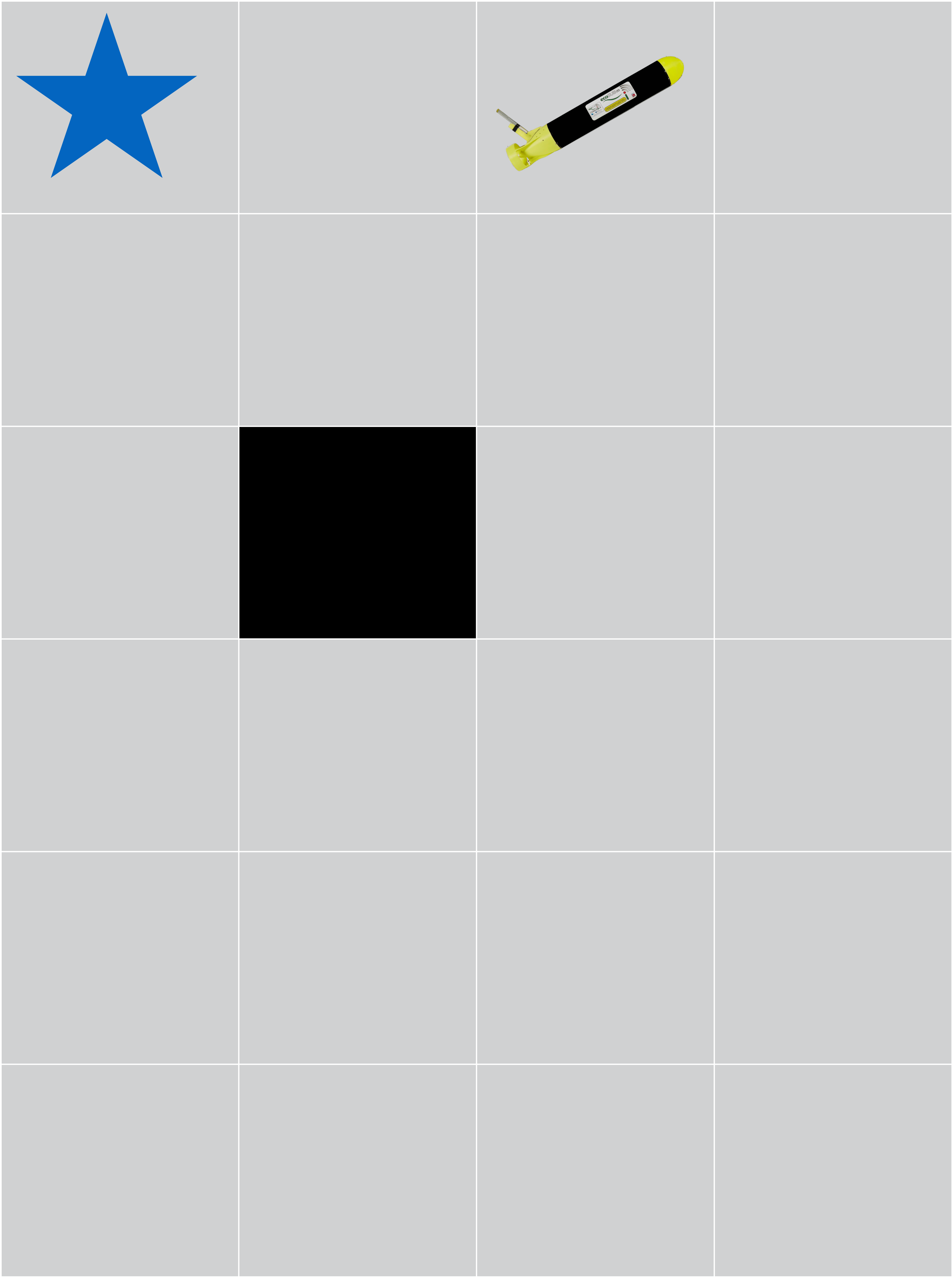
Example



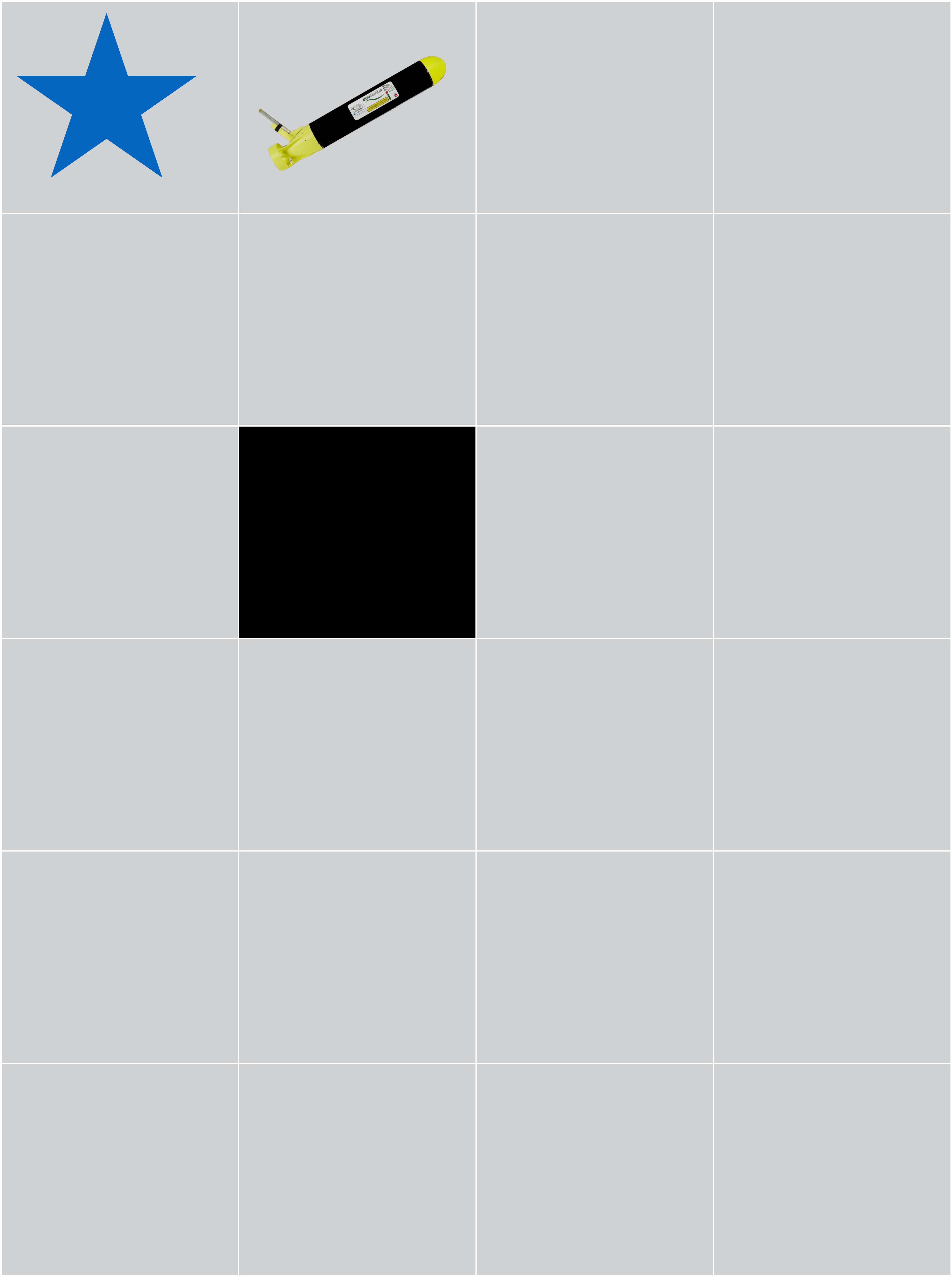
Example



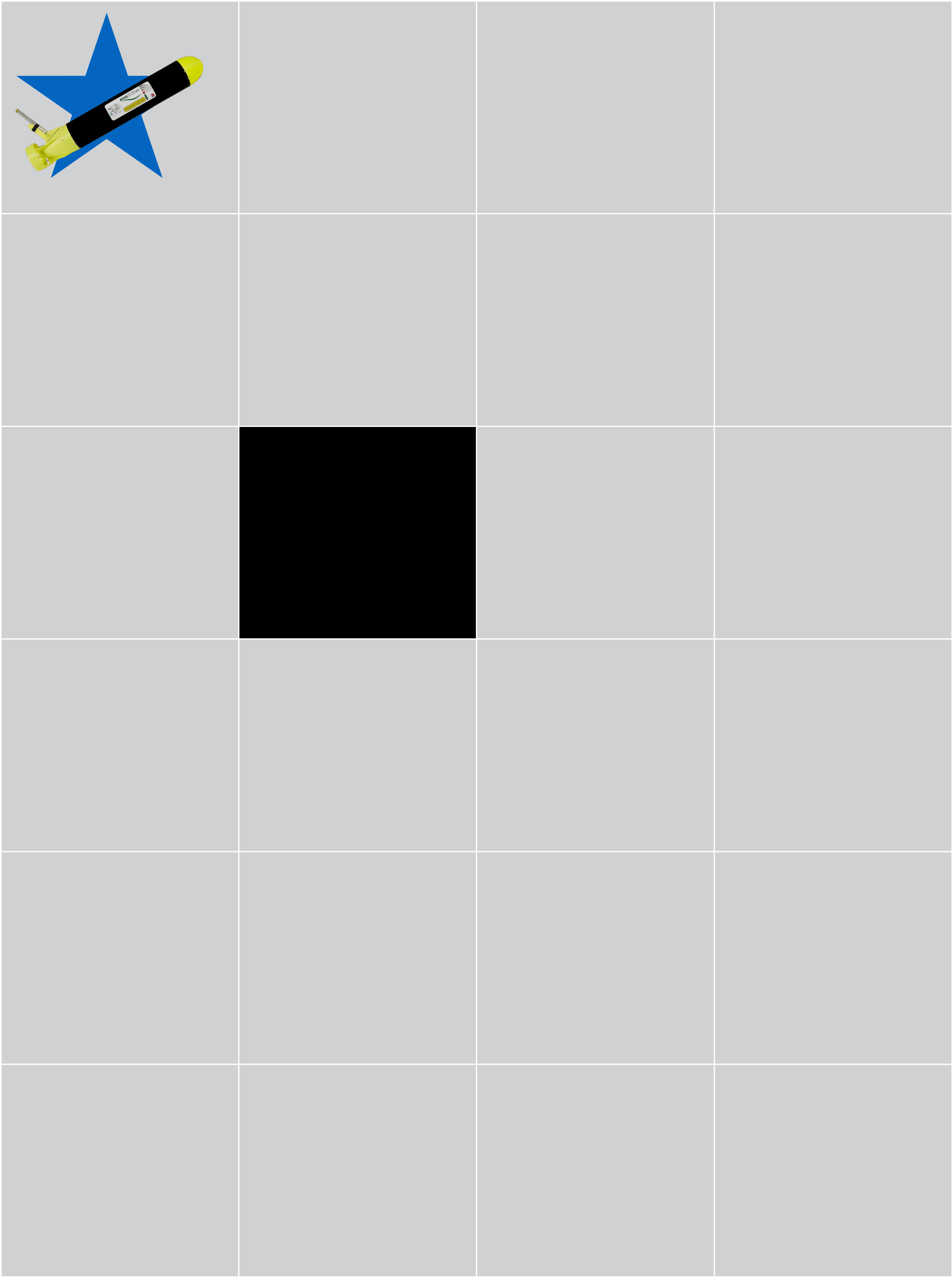
Example



Example

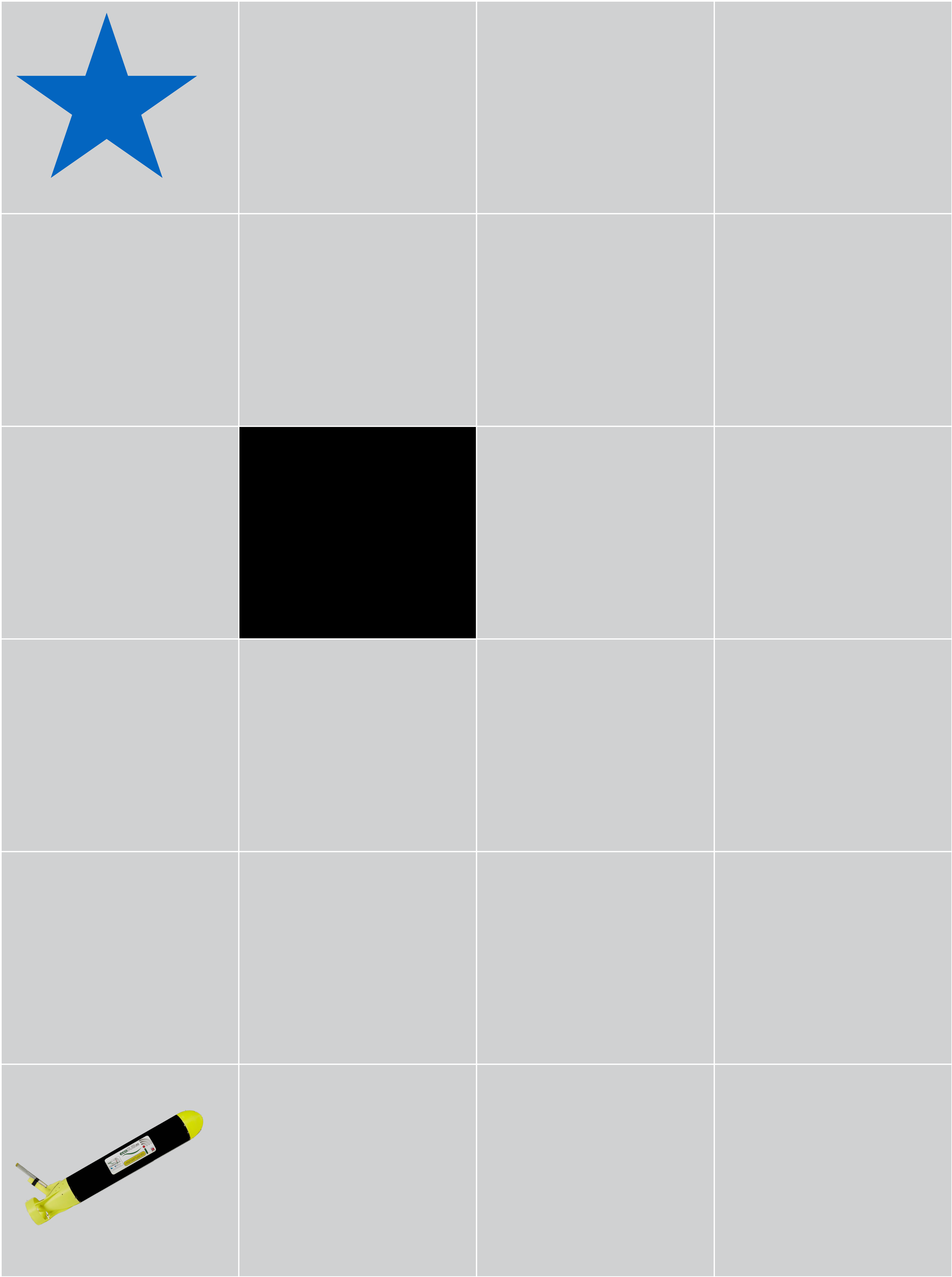


Example



Example

No disturbances - P_Z

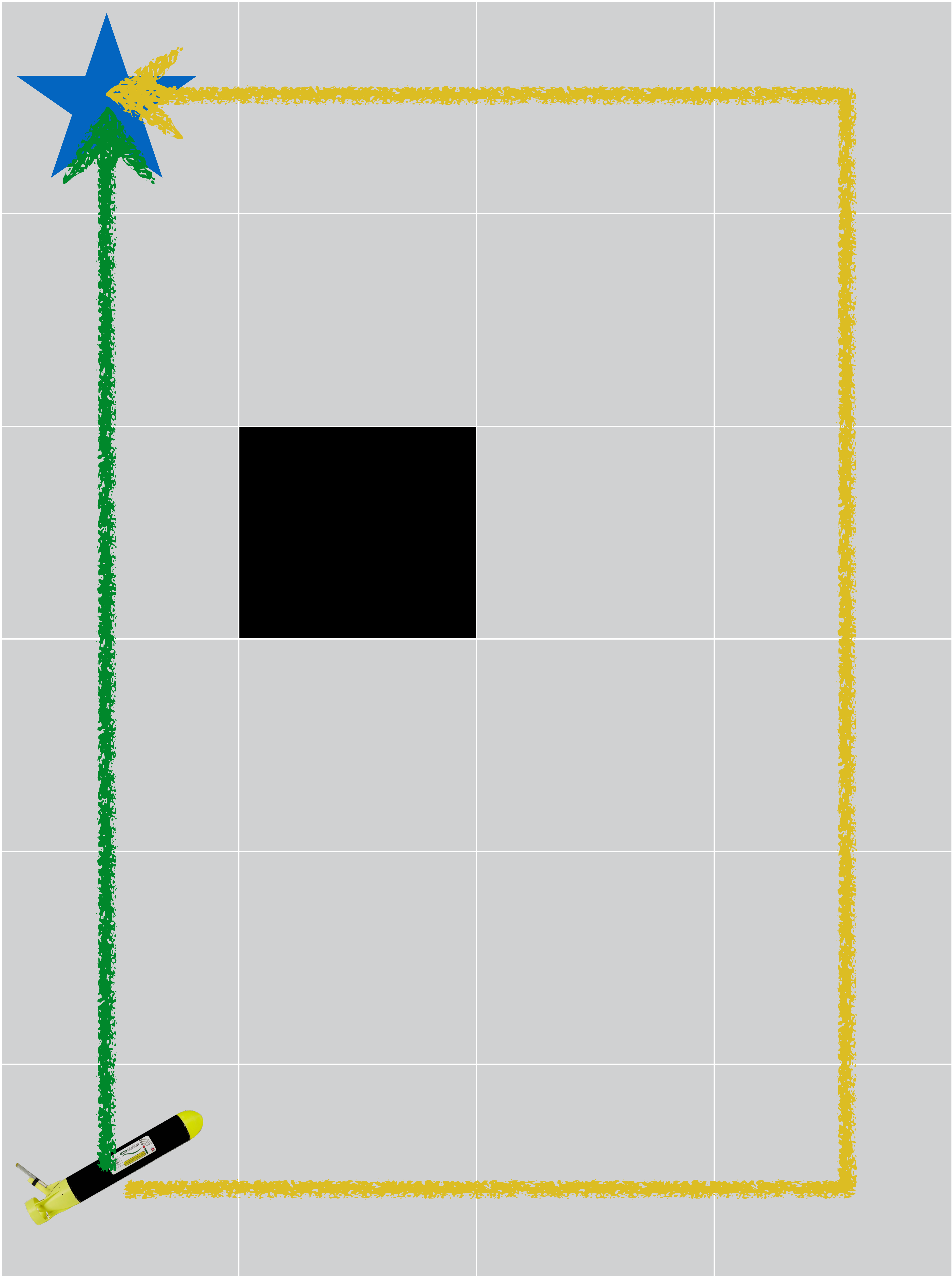


Example

No disturbances - P_Z

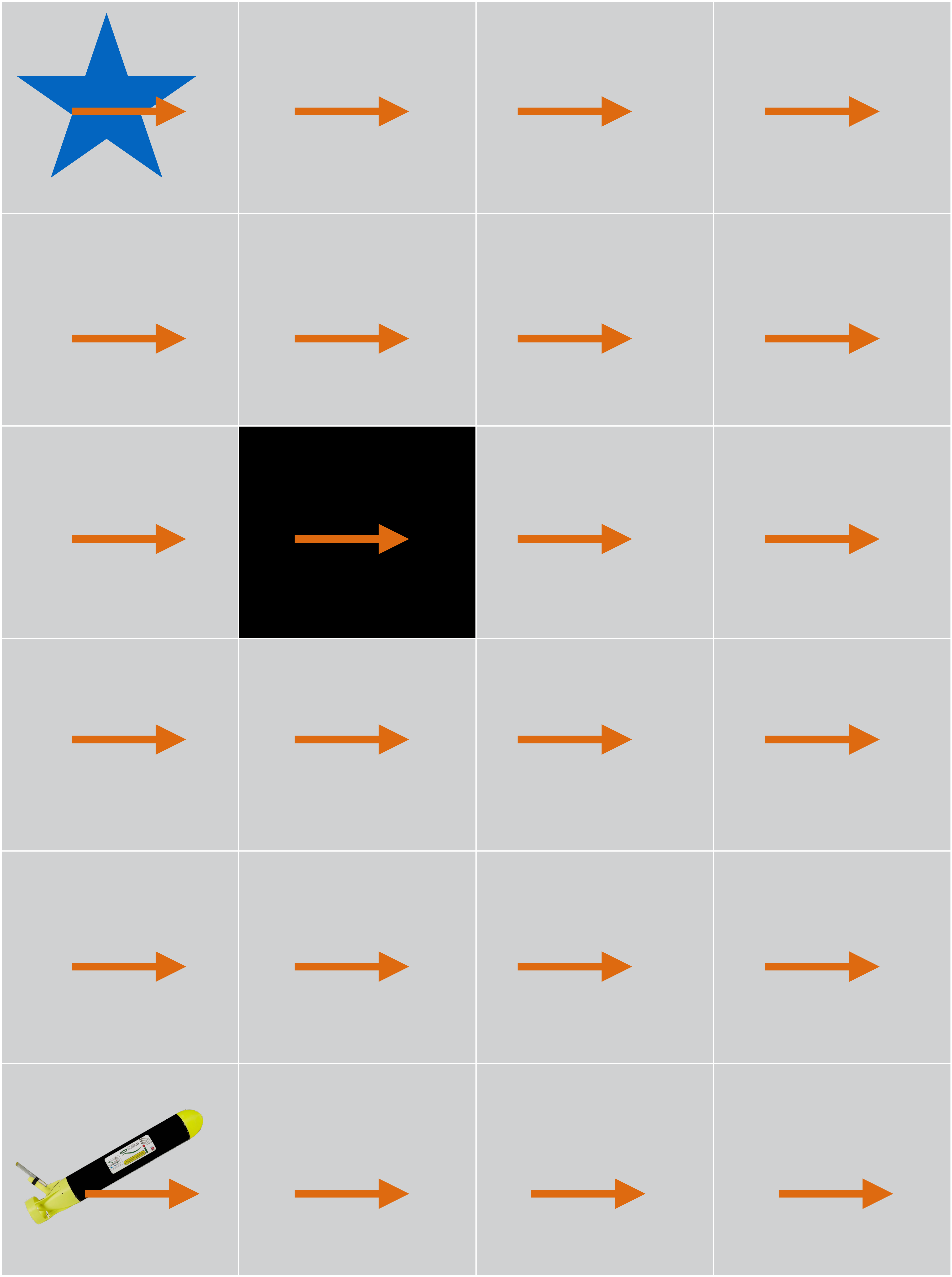
$V^{\pi_g, P_Z}(s_0) = 5$

$V^{\pi_y, P_Z}(s_0) = 11$



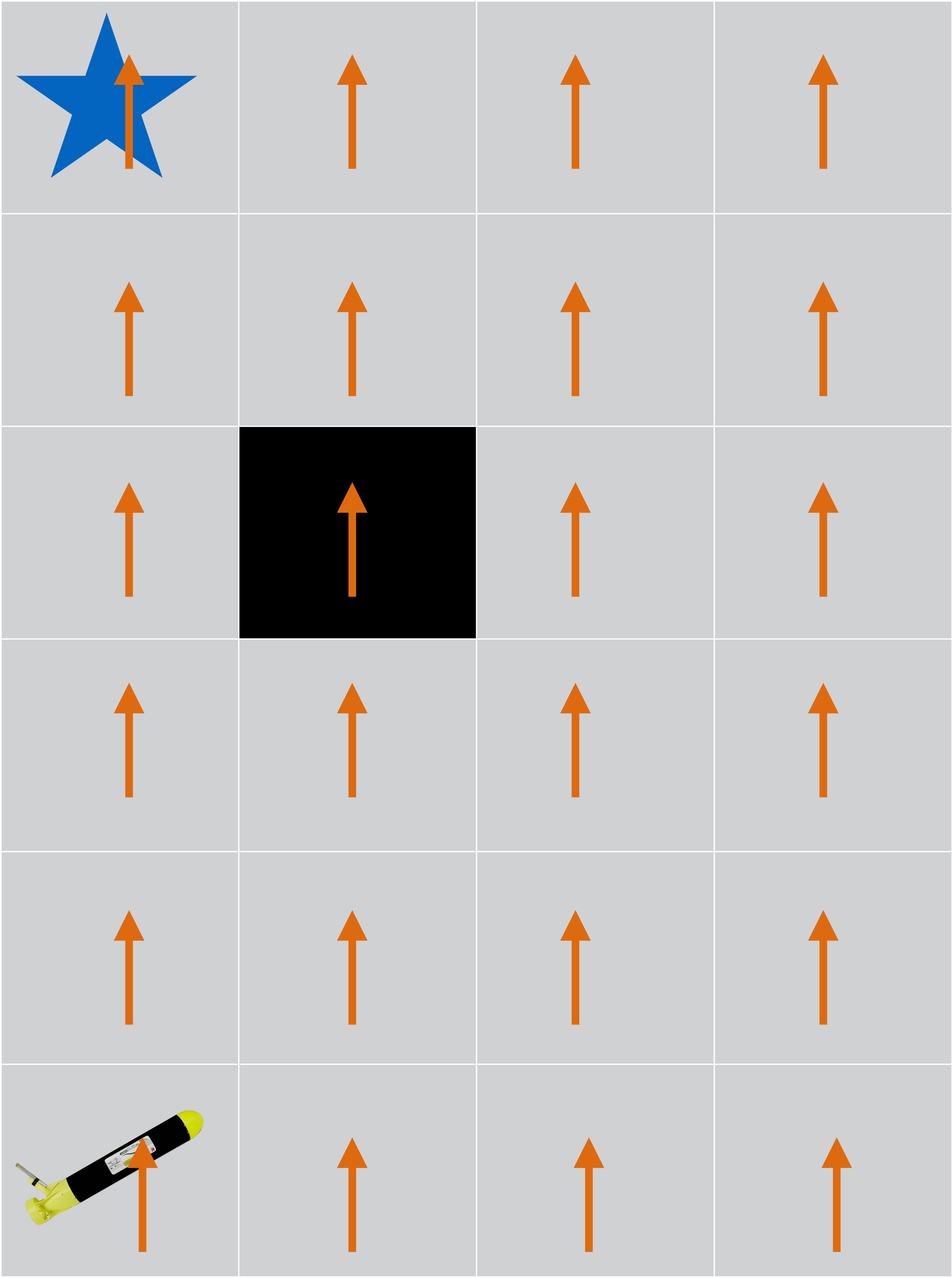
Example

East currents - P_E



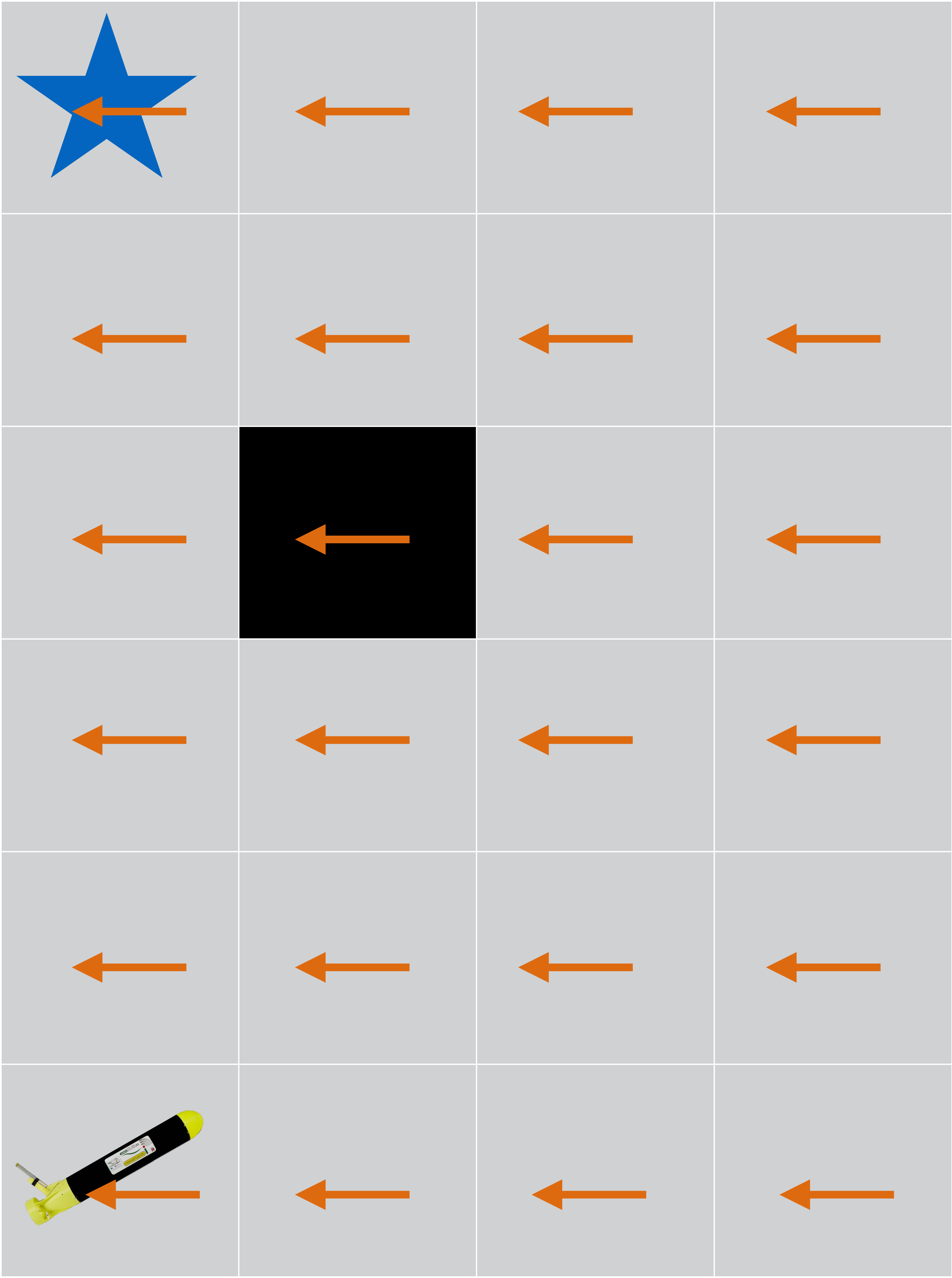
Example

North currents - P_N



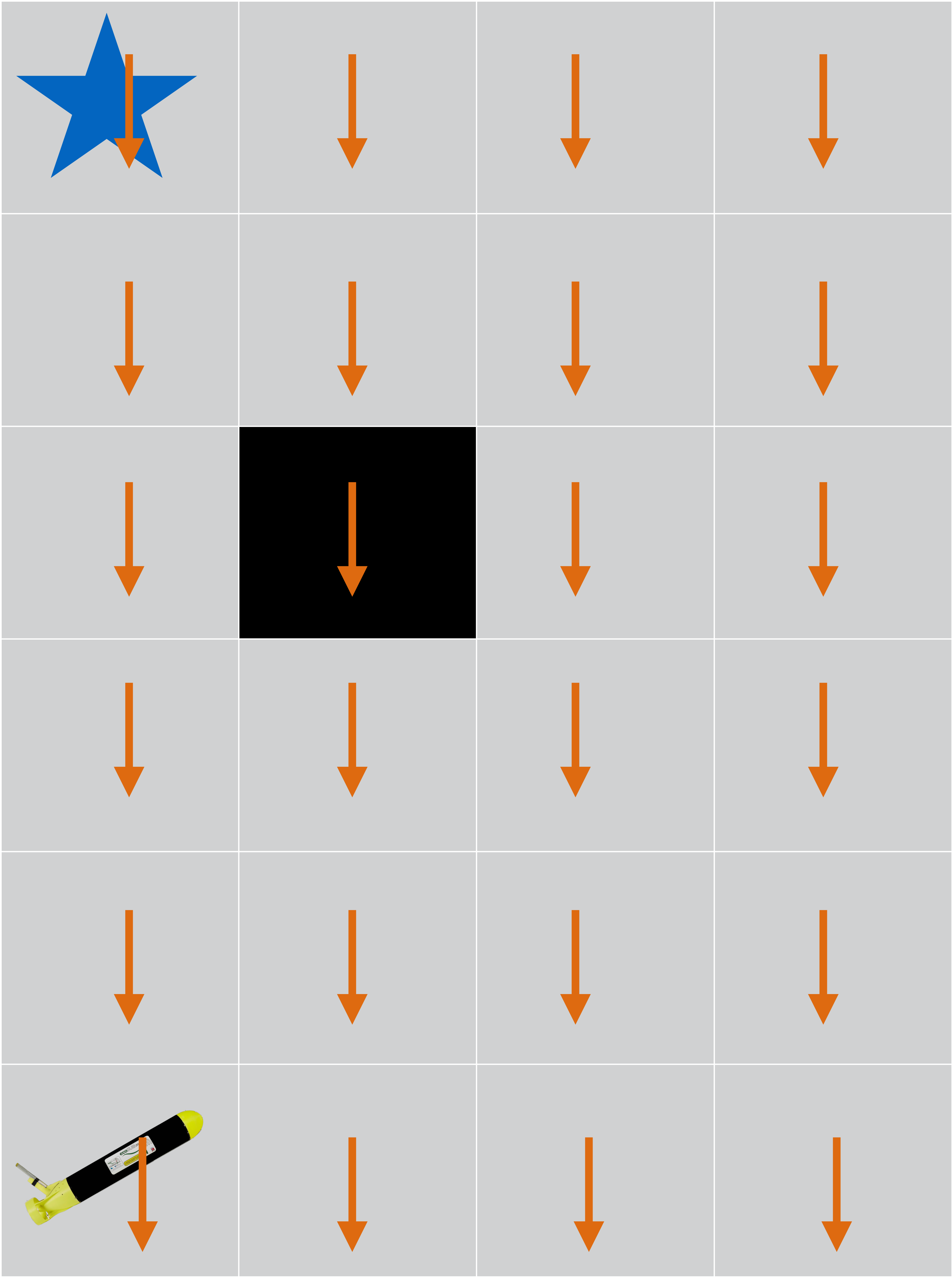
Example

West currents - P_W



Example

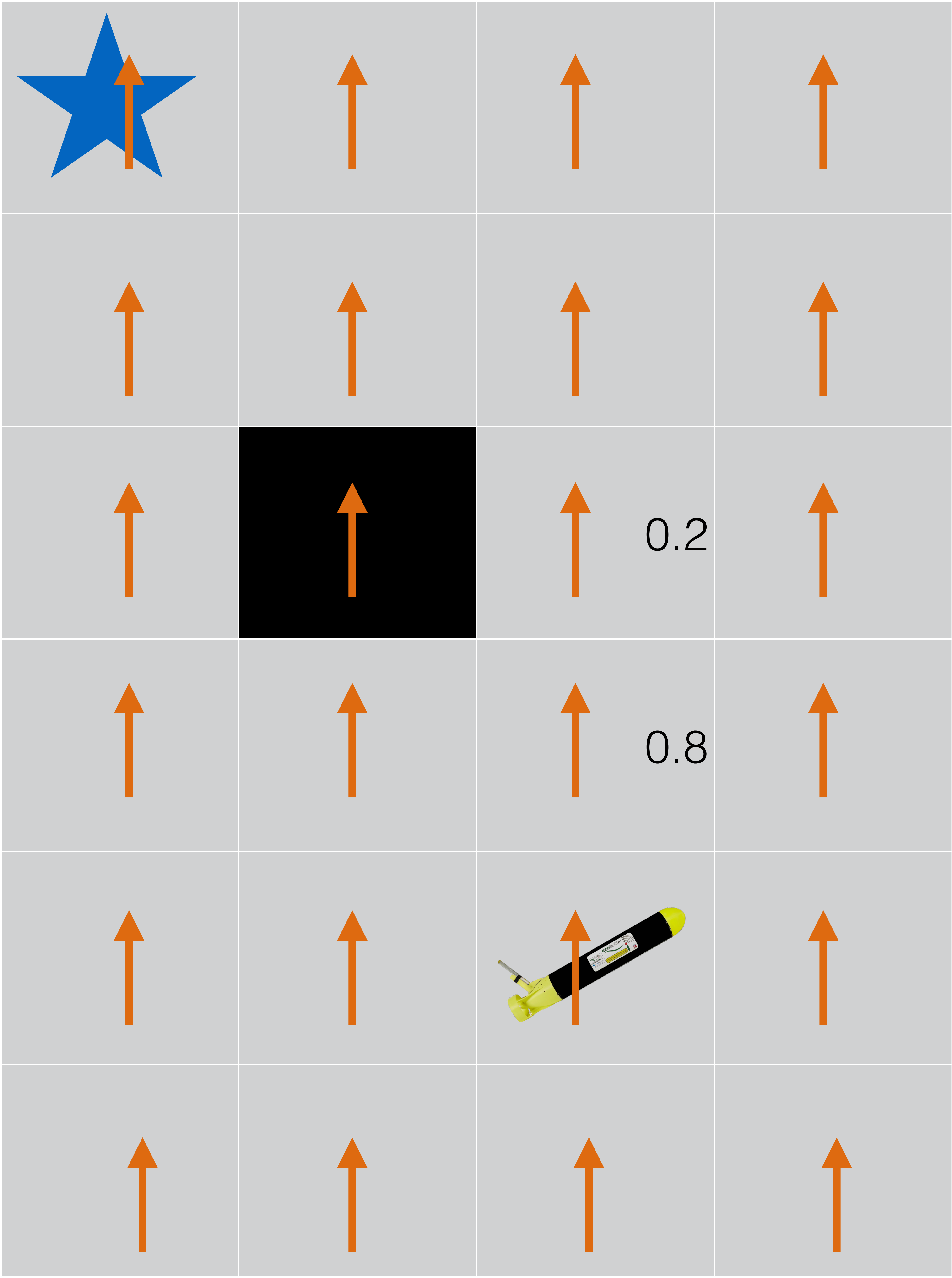
South currents - P_S



Example

North currents - P_N

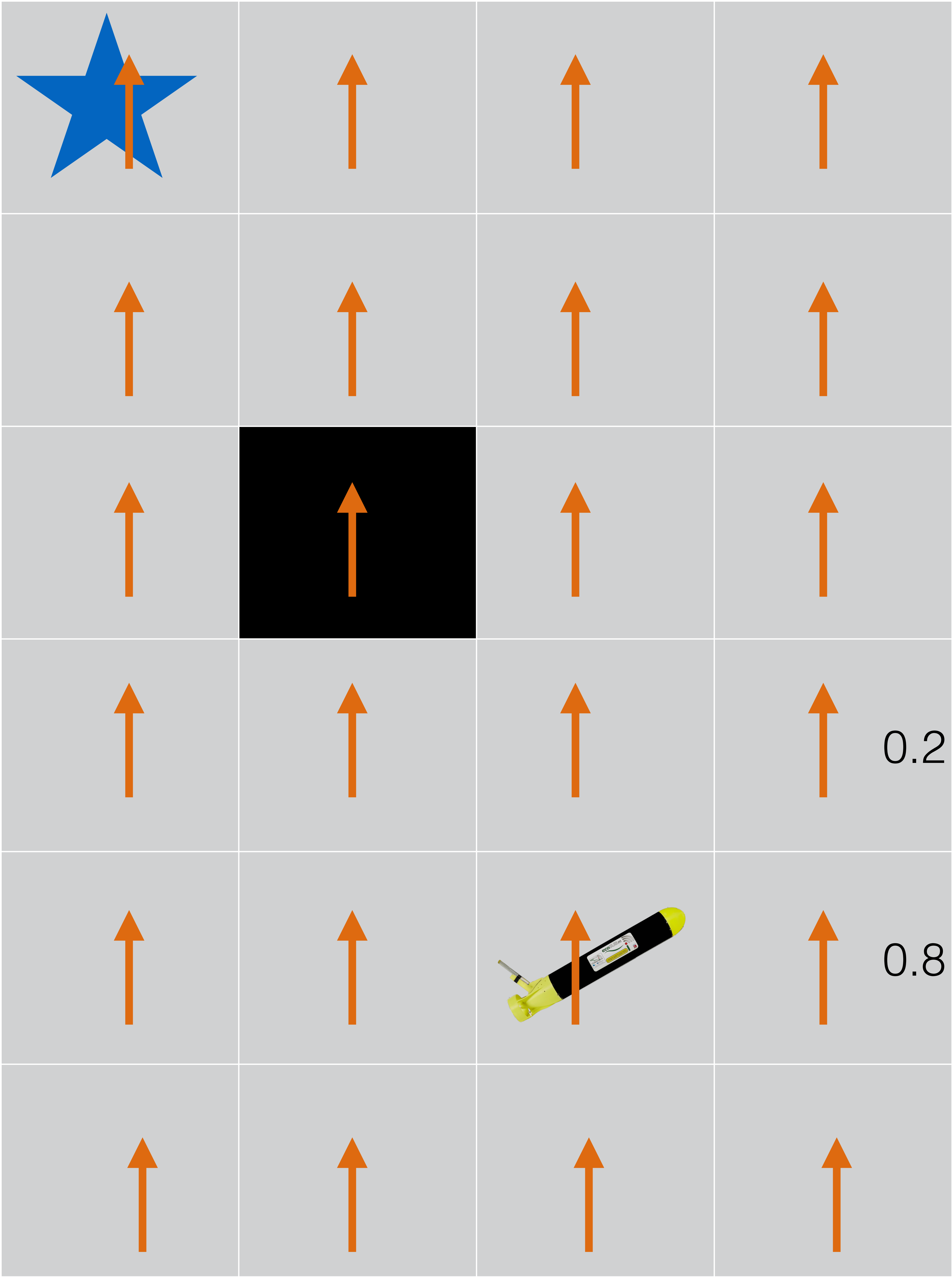
- Action: move up (N)



Example

North currents - P_N

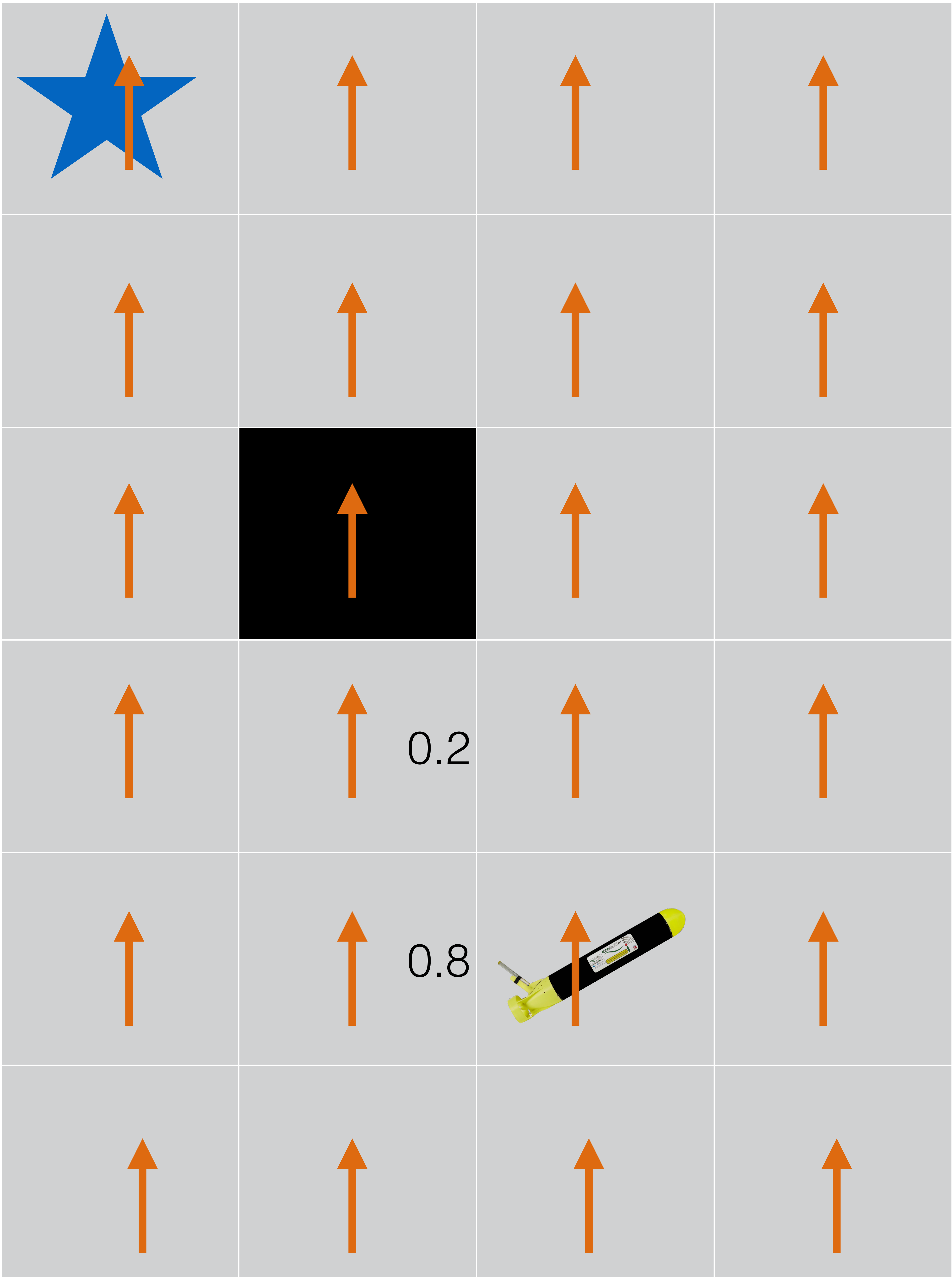
- Action: move east (E)



Example

North currents - P_N

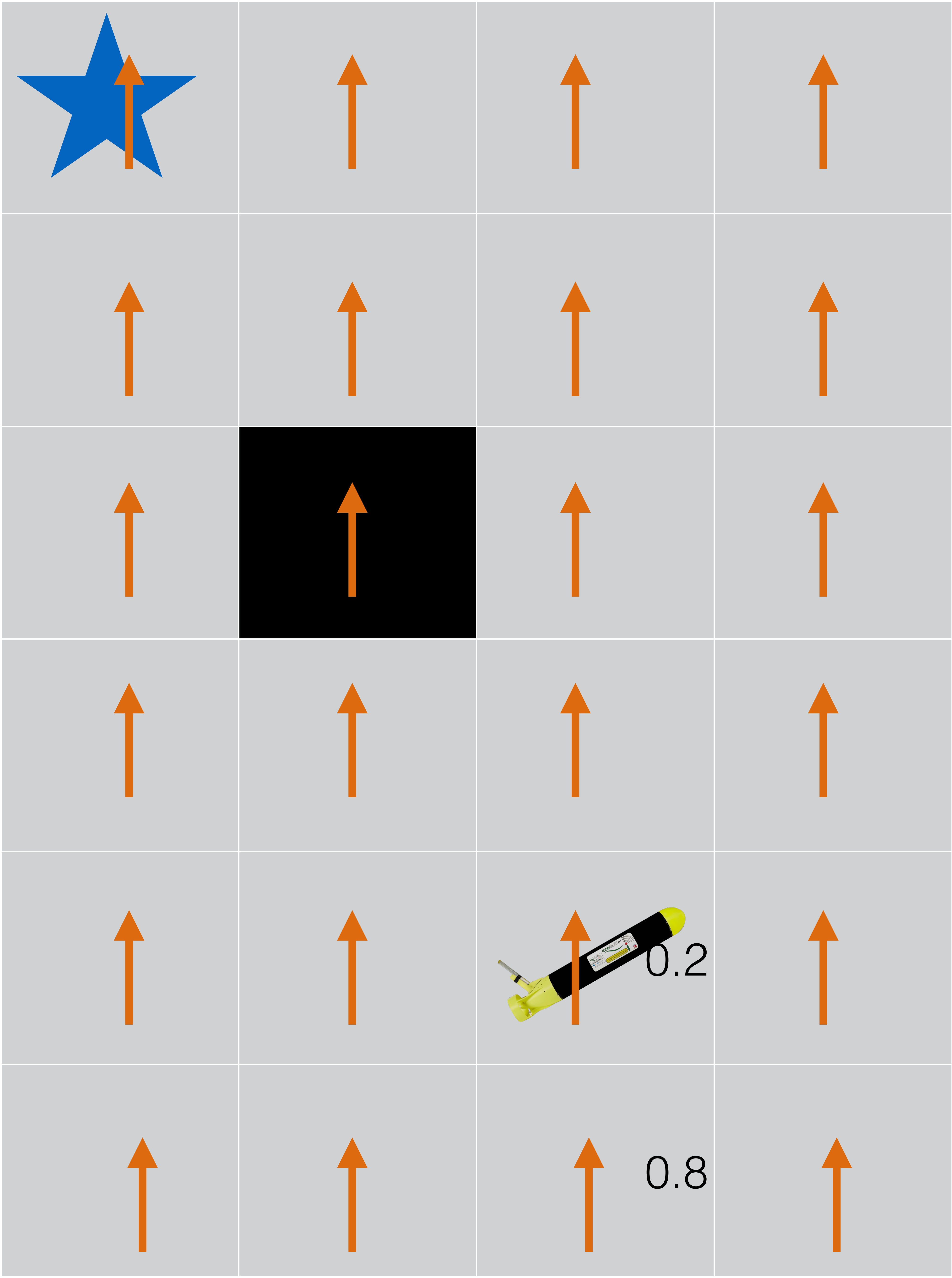
- Action: move west (W)



Example

North currents - P_S

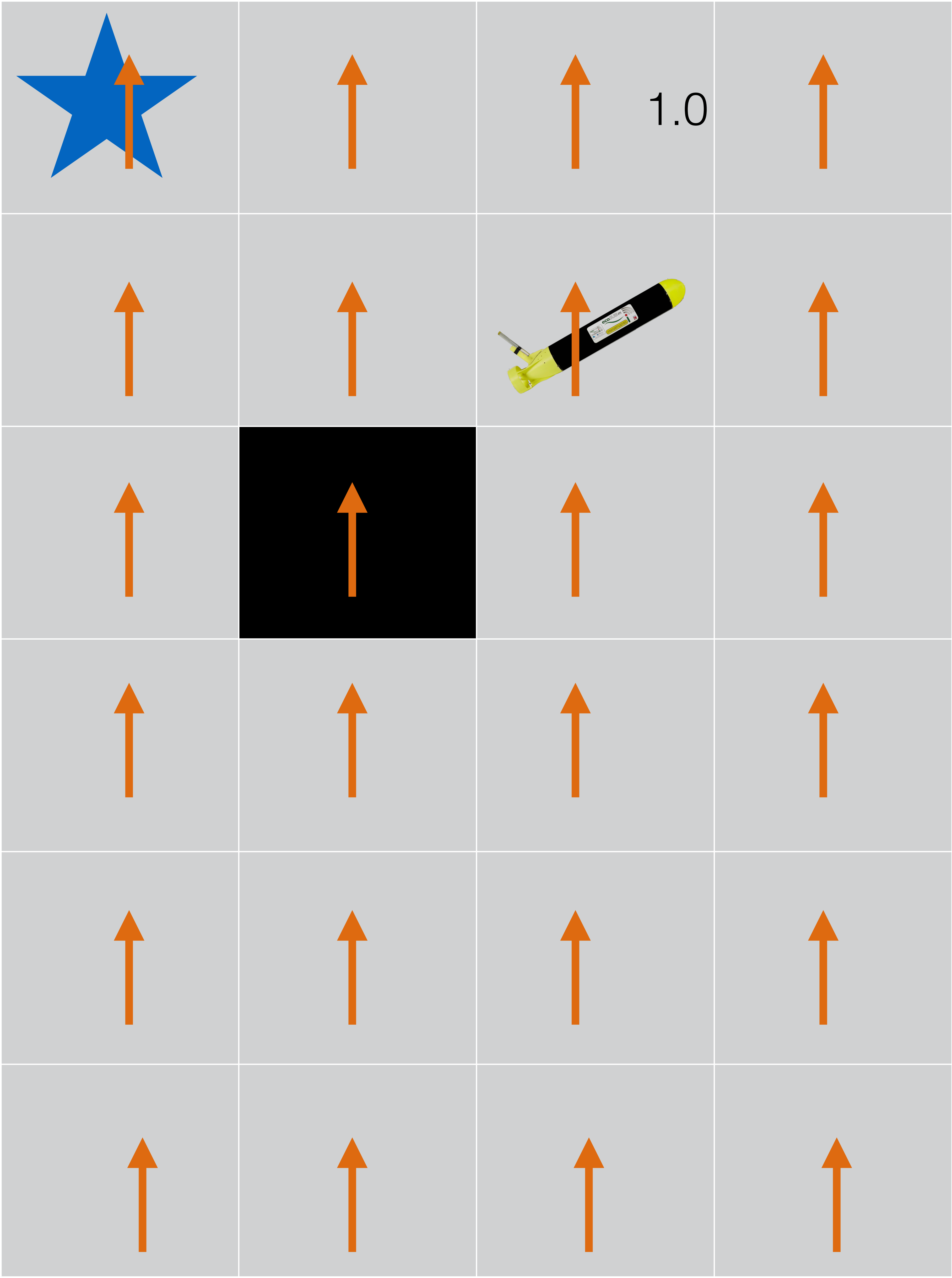
- Action: move south (S)



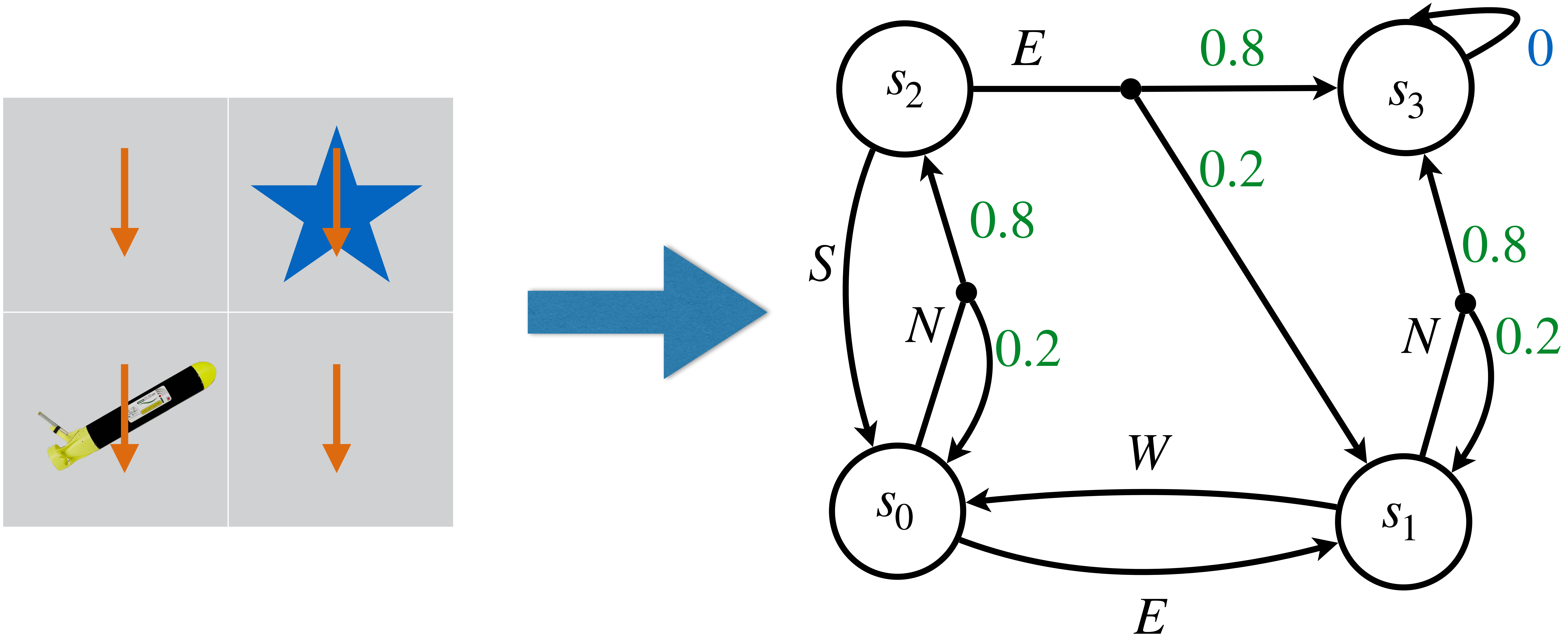
Example

North currents - P_N

- Action: move up (N)

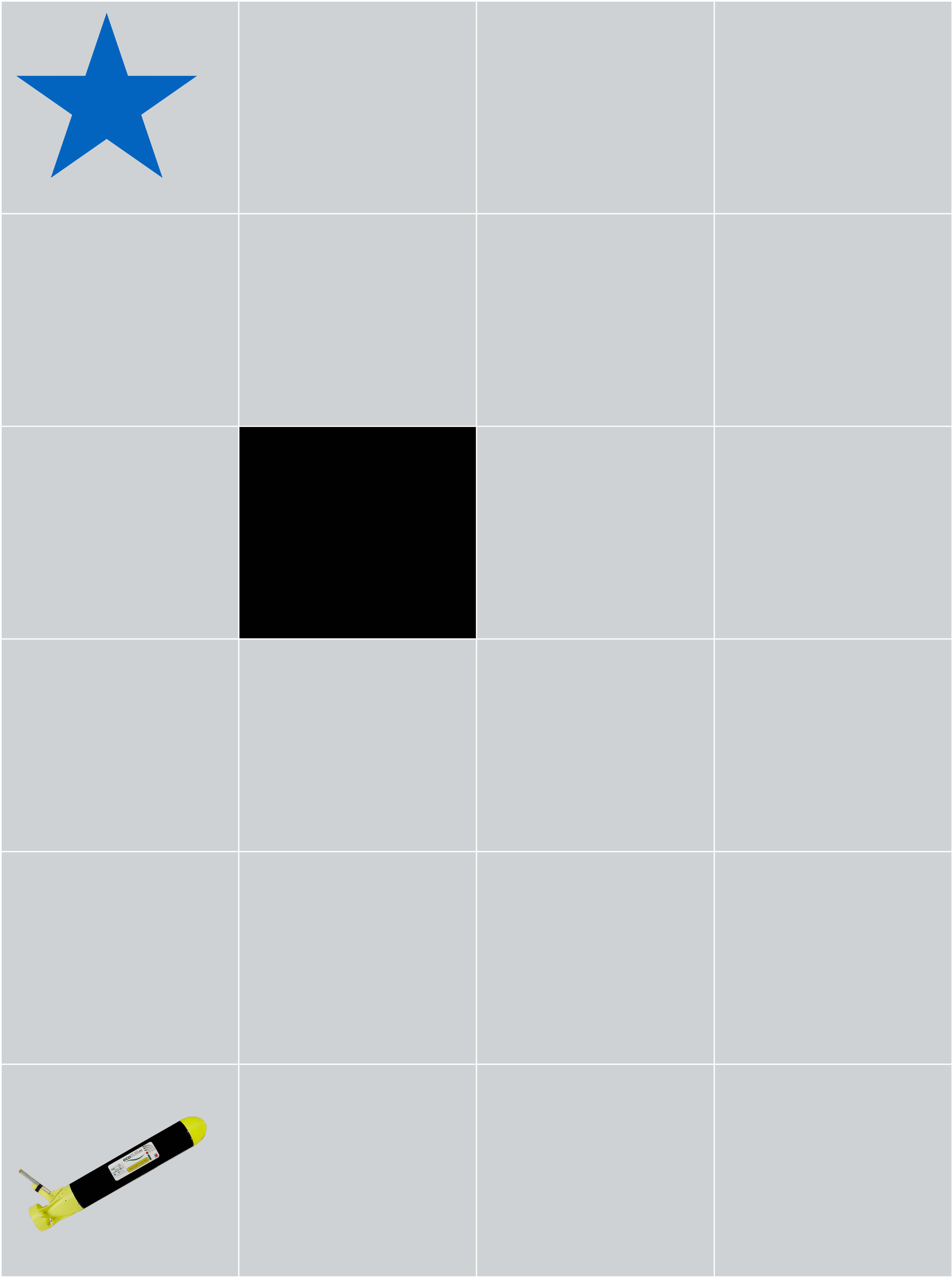


Example

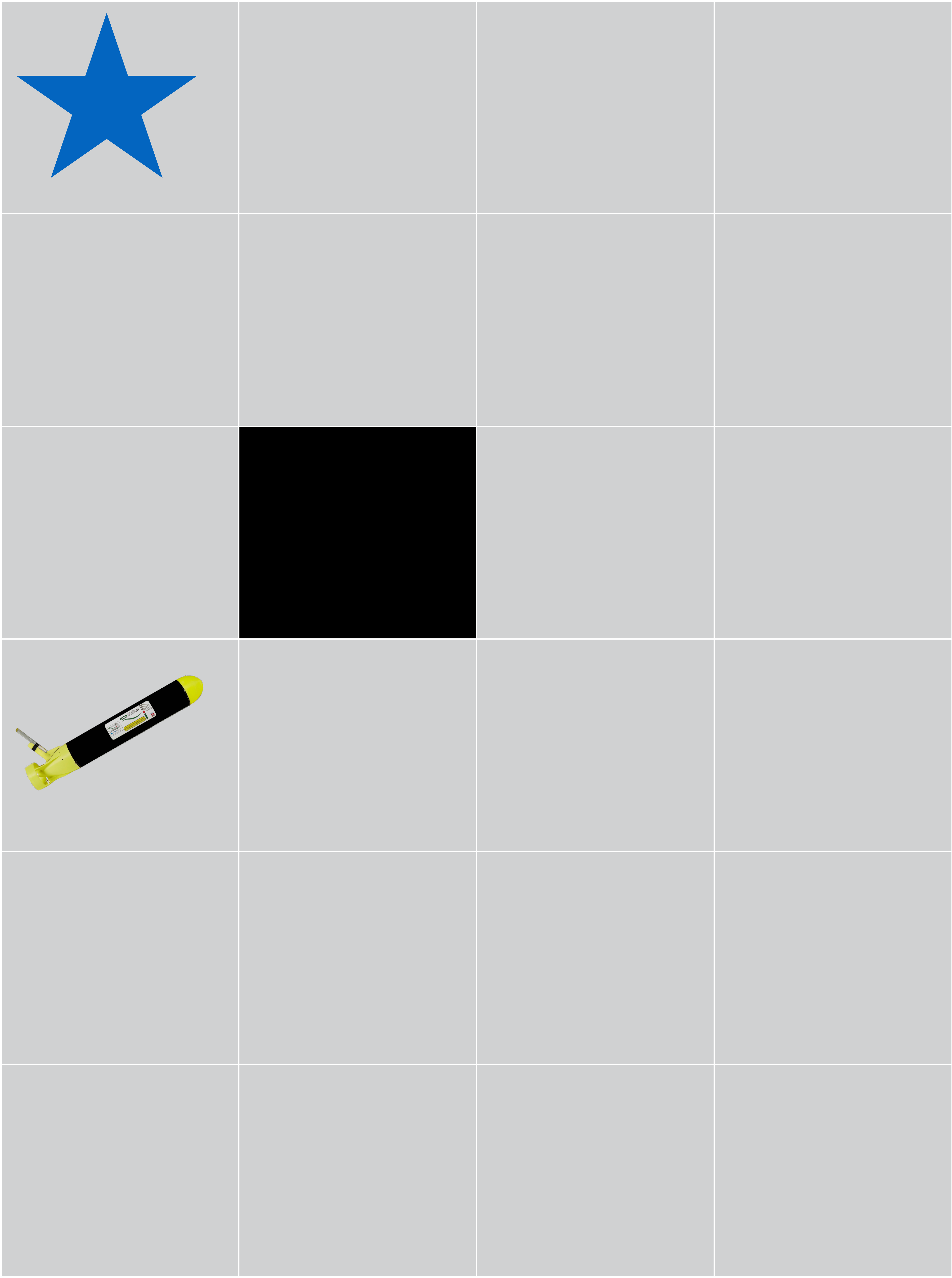


Example

$$\mathscr{P} = \{P_Z, P_N, P_S, P_E, P_W\}$$

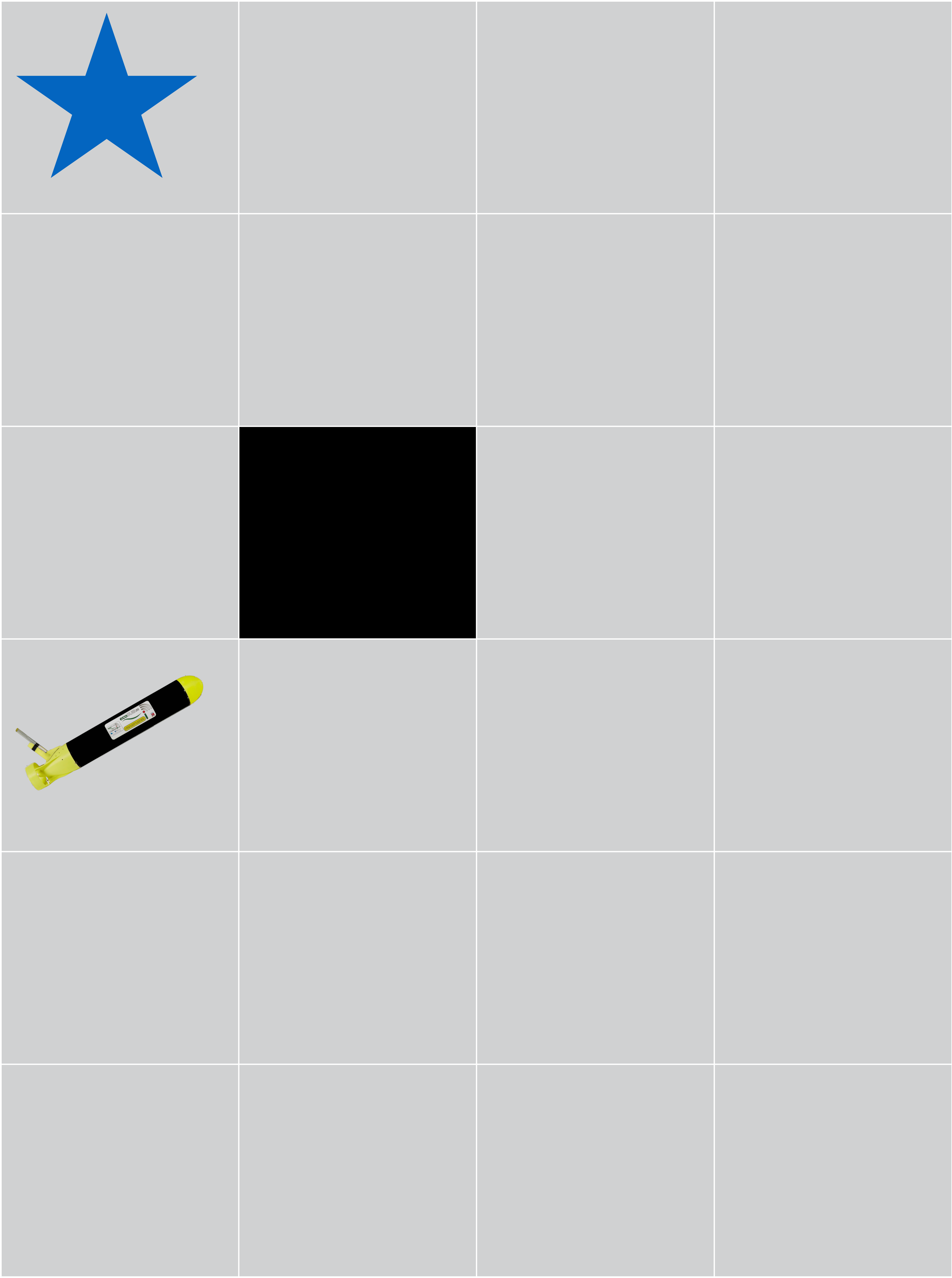


Example



Example

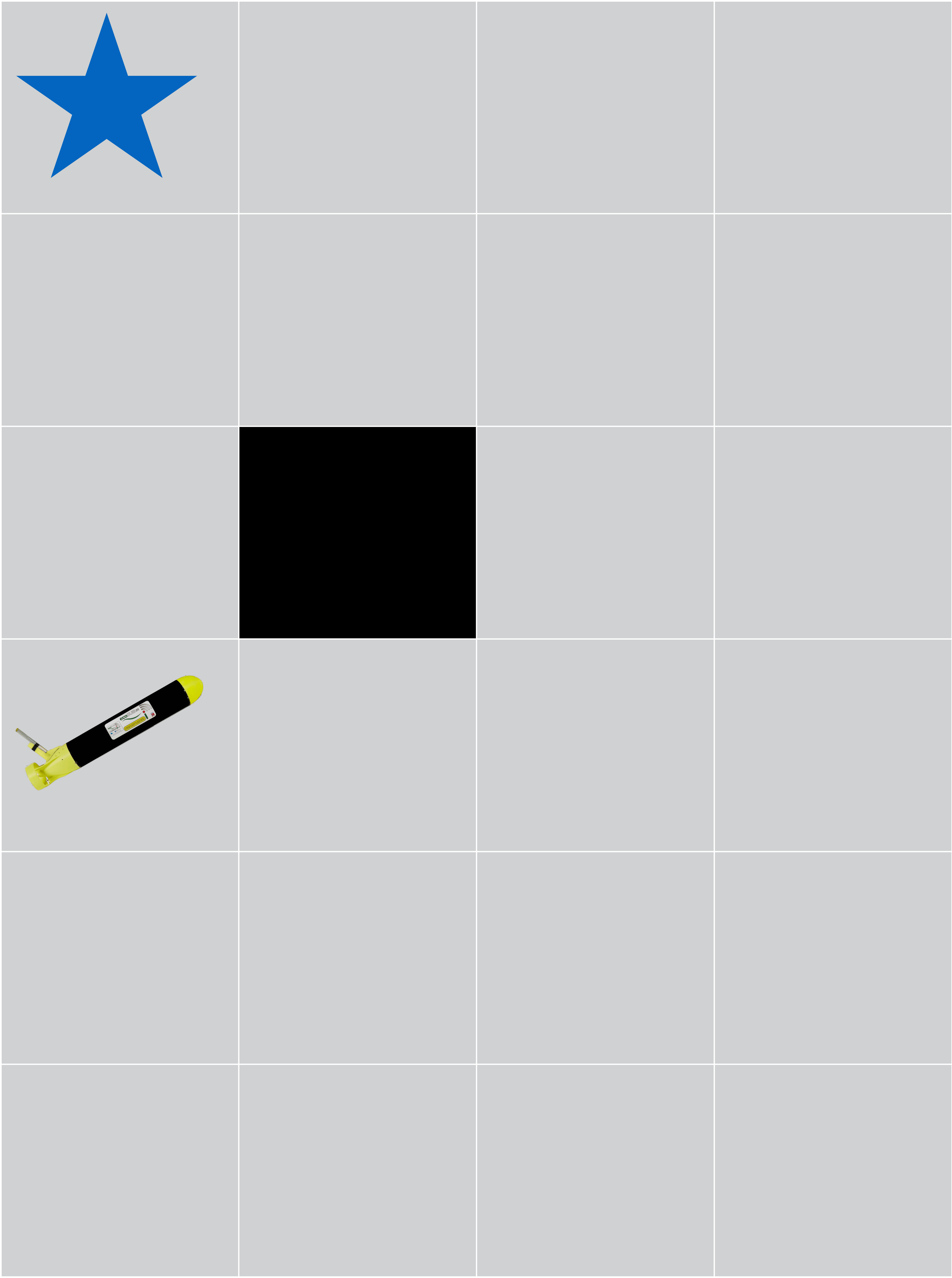
- Rectangularity assumption means environment can choose the most damaging current for every state-action pair



Example

- Rectangularity assumption means environment can choose the most damaging current for every state-action pair

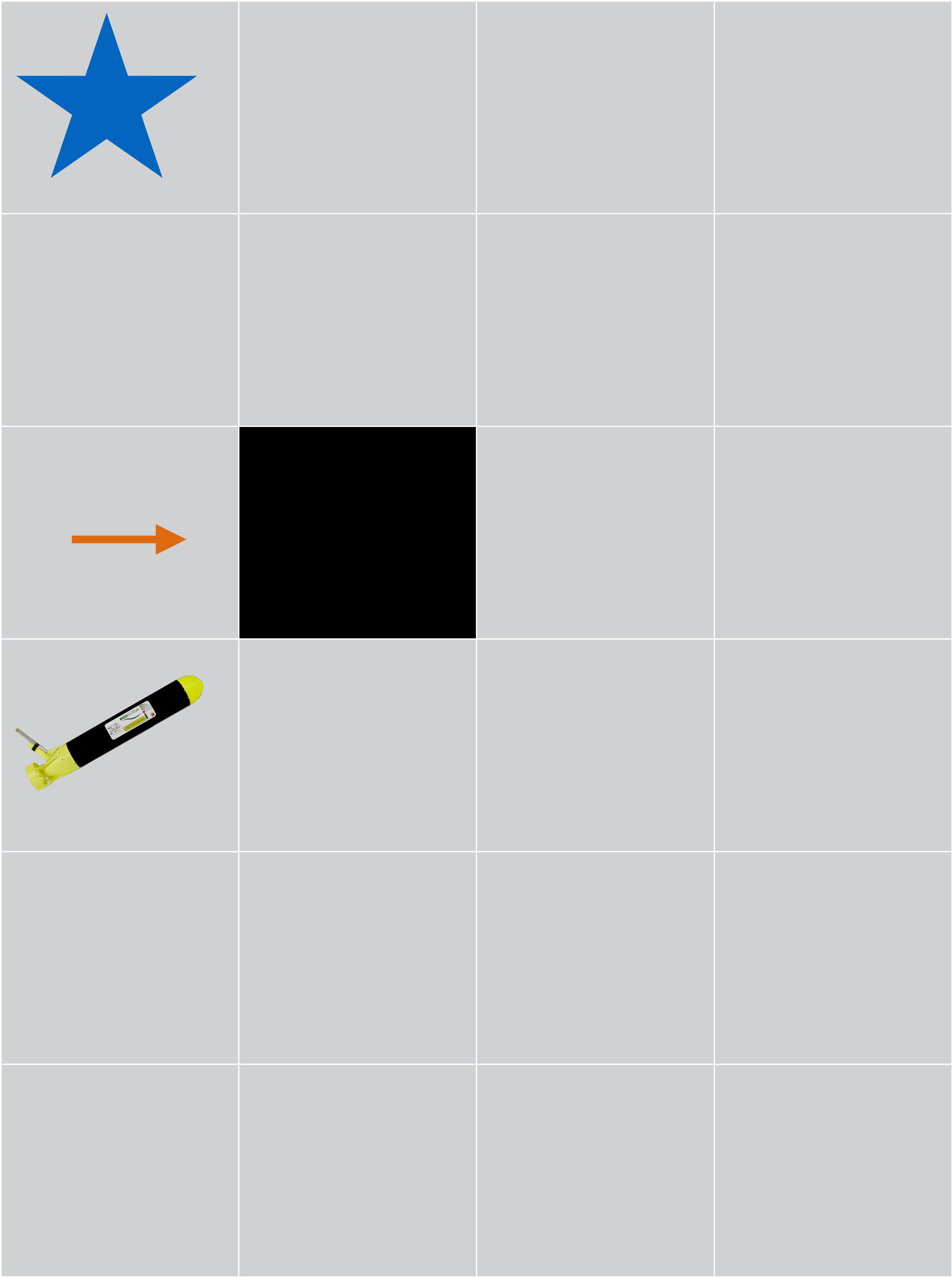
- Action: move up (N)



Example

- Rectangularity assumption means environment can choose the most damaging current for every state-action pair

- Action: move up (N)

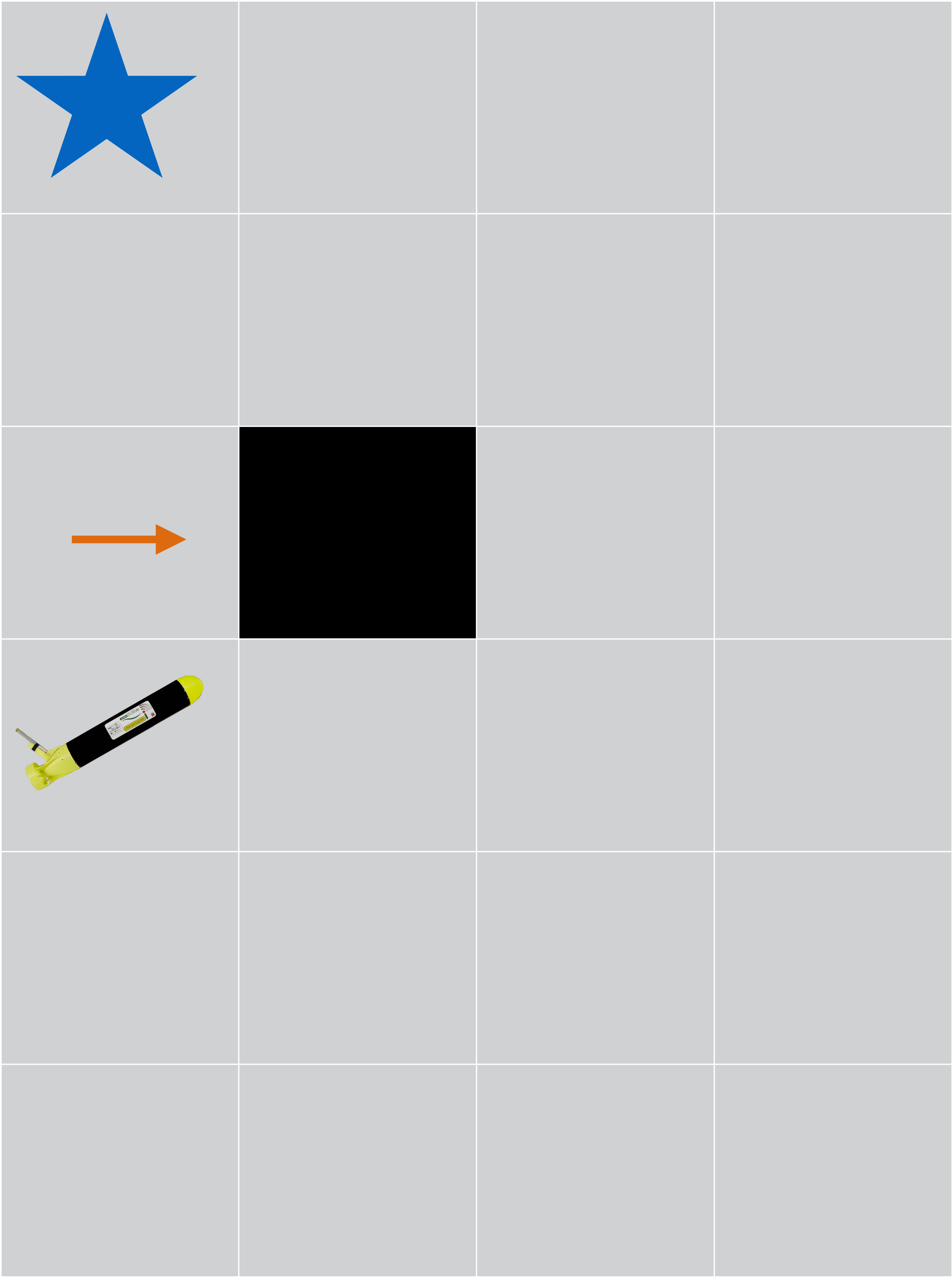


Example

- Rectangularity assumption means environment can choose the most damaging current for every state-action pair

- Action: move up (N)

- Action: move east (E)

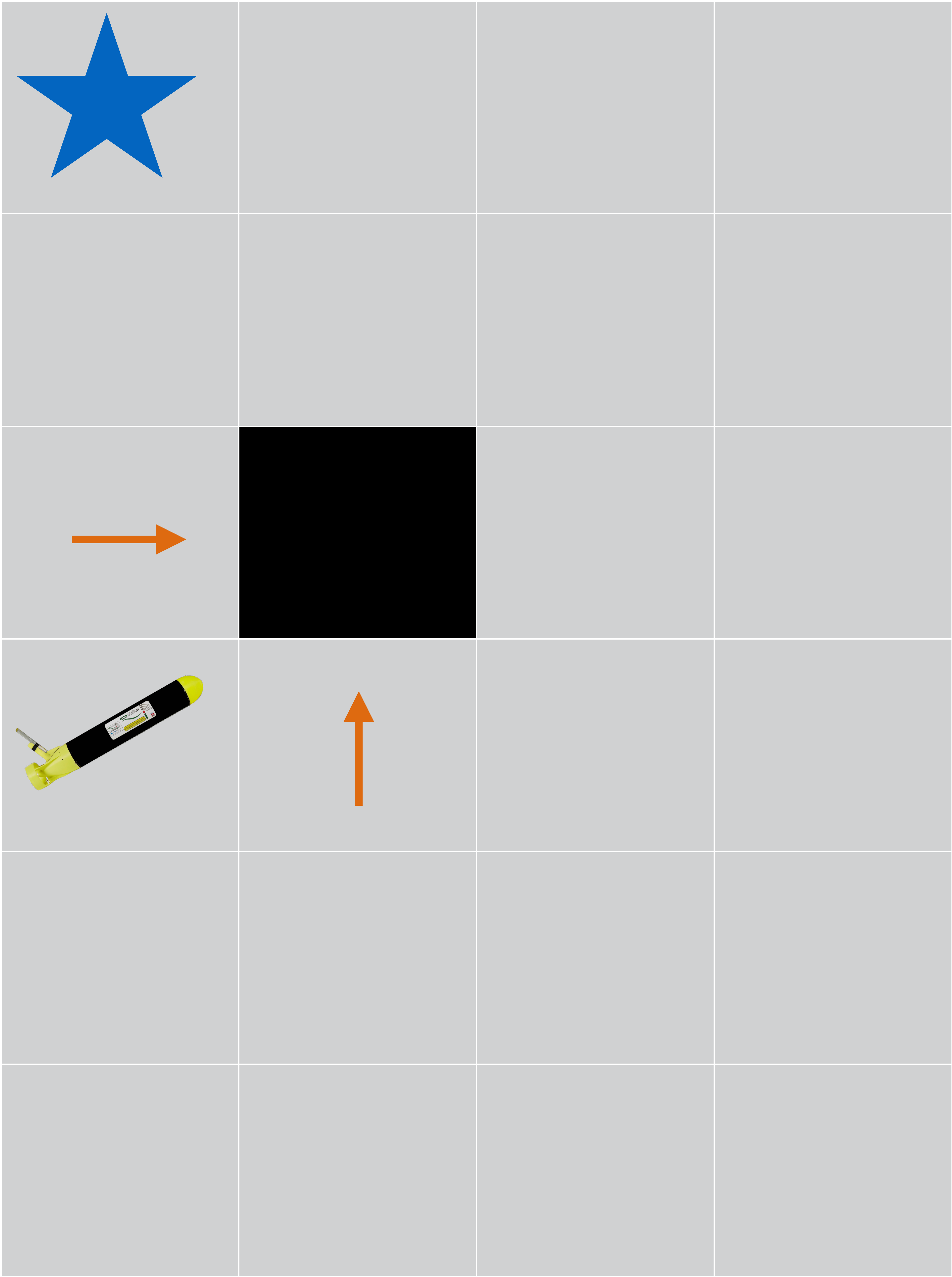


Example

- Rectangularity assumption means environment can choose the most damaging current for every state-action pair

- Action: move up (N)

- Action: move east (E)

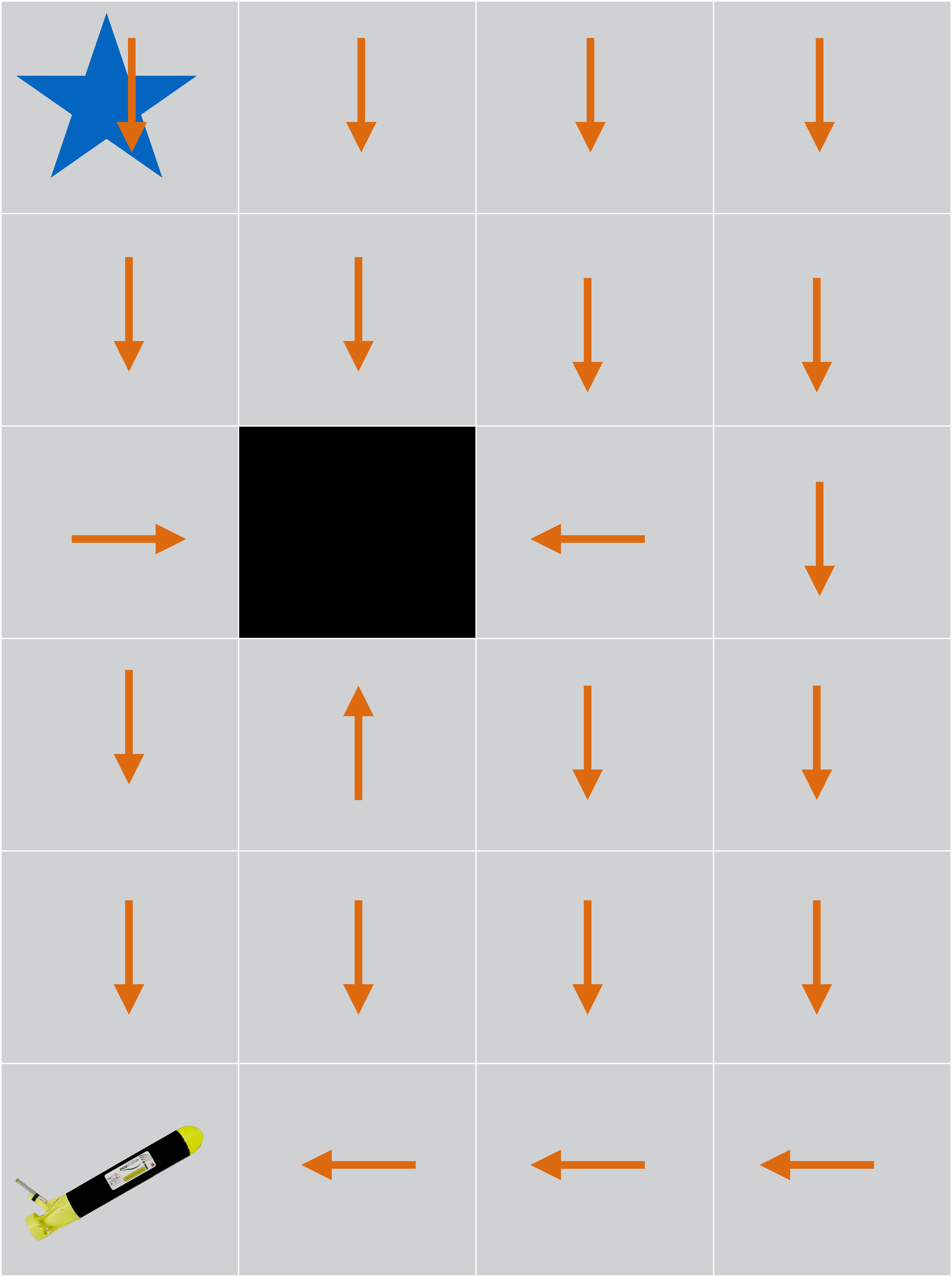


Example

- Rectangularity assumption means environment can choose the most damaging current for every state-action pair

- Action: move up (N)

- Action: move east (E)



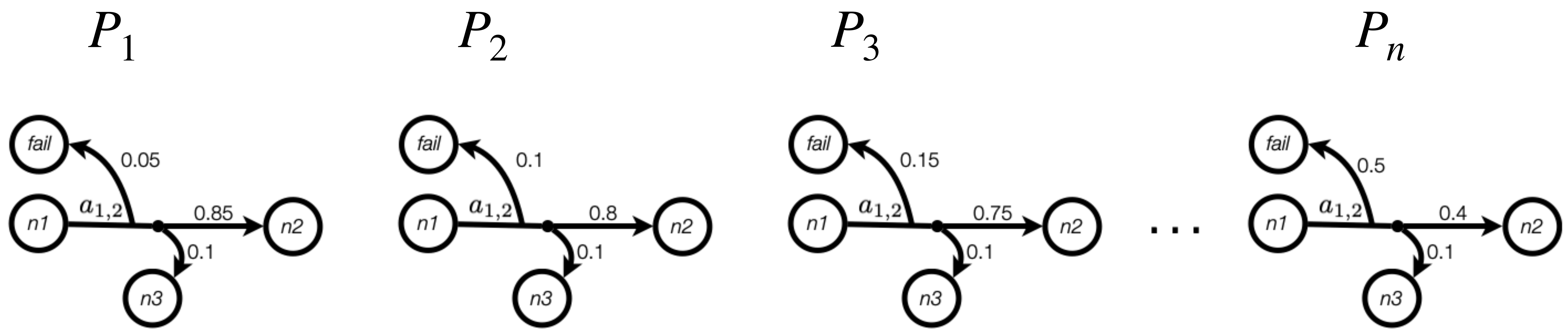
Sample-based UMDPs

We will consider SSP MDPs

$\mathscr{M} = (S, s_0, A, \mathscr{P}, C, goal)$

No rectangularity assumption

$\mathscr{P} = \{P_1, \ldots, P_n\}$ where $P_i : S \times A \rightarrow Dist(S)$



Considering dependencies

- Considering dependencies reduces conservativeness of solutions
- It also enables adaptivity to environment conditions
  - If currents are pushing me north, then I can navigate west of the high cost area
  - Optimal policies are typically finite-memory and randomised
  - Solution approaches are typically NP-hard
- We will look into approximate solutions from now on

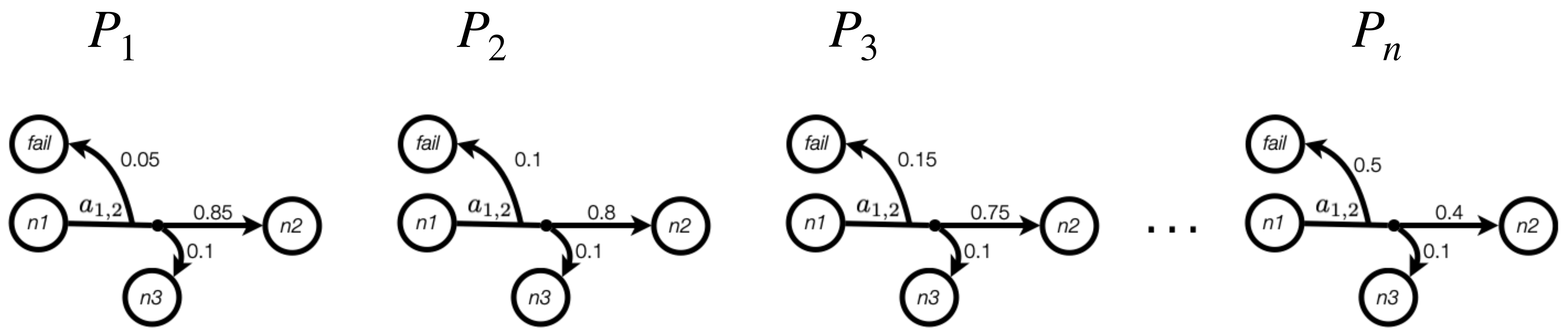
Worst-case optimisation

$$V^{wc}(s) = \min_{\pi \in \Pi} \max_{P \in \{P_1,\ldots,P_n\}} V^{\pi,P}(s)$$

$$\mathcal{M} = (S, s_0, A, \mathcal{P}, C, goal)$$

$\mathcal{P} = \{P_1, \ldots, P_n\}$ where $P_i : S \times A \to Dist(S)$

- Solutions can be too conservative
- Requires finite-memory randomised policies
- NP-hard
- Not well studied



Regret optimisation

$$V^{*,P}(s) = \text{argmin}_{\pi \in \Pi} V^{\pi,P}(s)$$

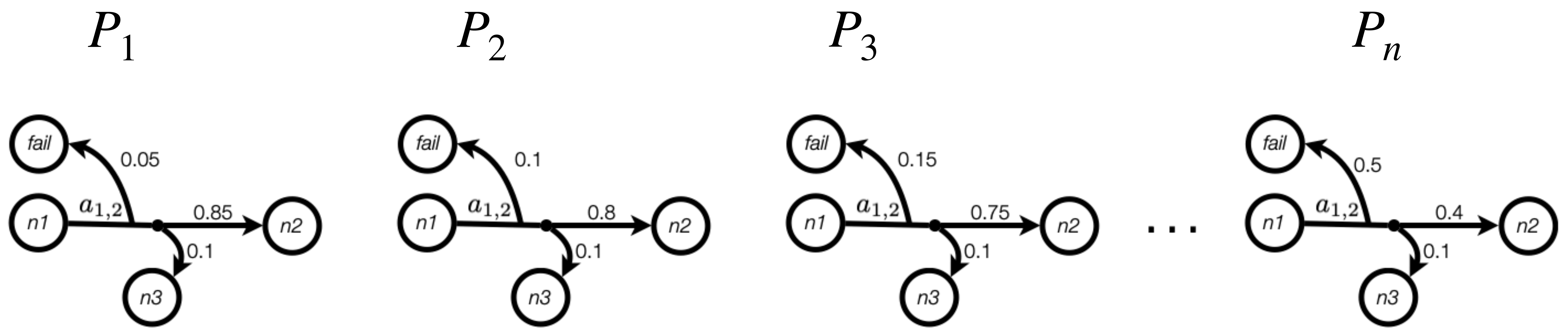
$$reg^{\pi,P}(s) = V^{\pi,P}(s) - V^{*,P}(s)$$

$$V^{reg}(s) = \min_{\pi \in \Pi} \max_{P \in \mathscr{P}} reg^{\pi,P}(s)$$

$$\mathscr{M} = (S, s_0, A, \mathscr{P}, C, goal)$$

$\mathscr{P} = \{P_1, \ldots, P_n\}$ where $P_i : S \times A \rightarrow Dist(S)$

- Requires finite-memory randomised policies
- NP-hard

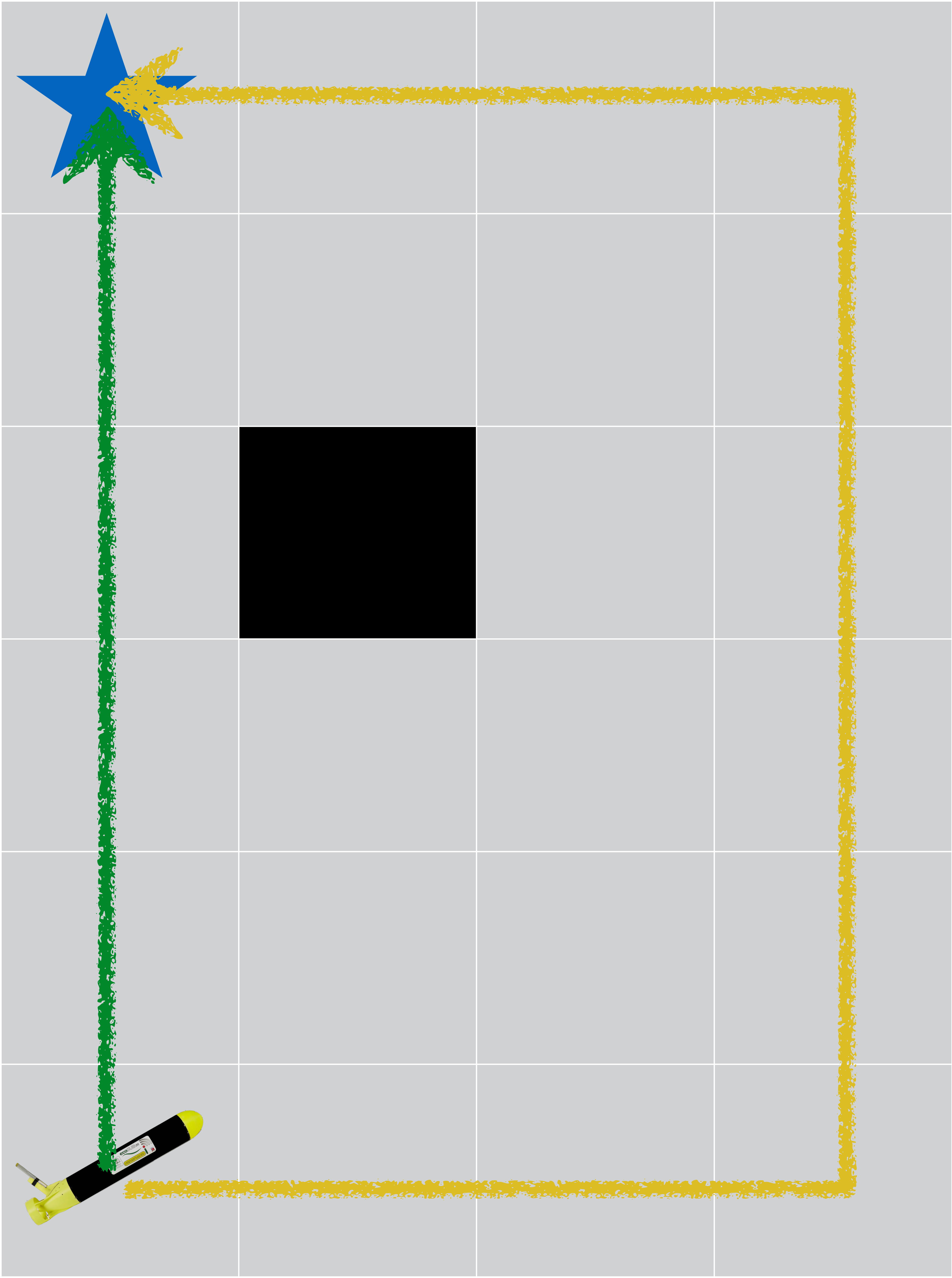


Example

No disturbances - P_Z

$V^{\pi_g, P_Z}(s_0) = 5$

$V^{\pi_y, P_Z}(s_0) = 11$

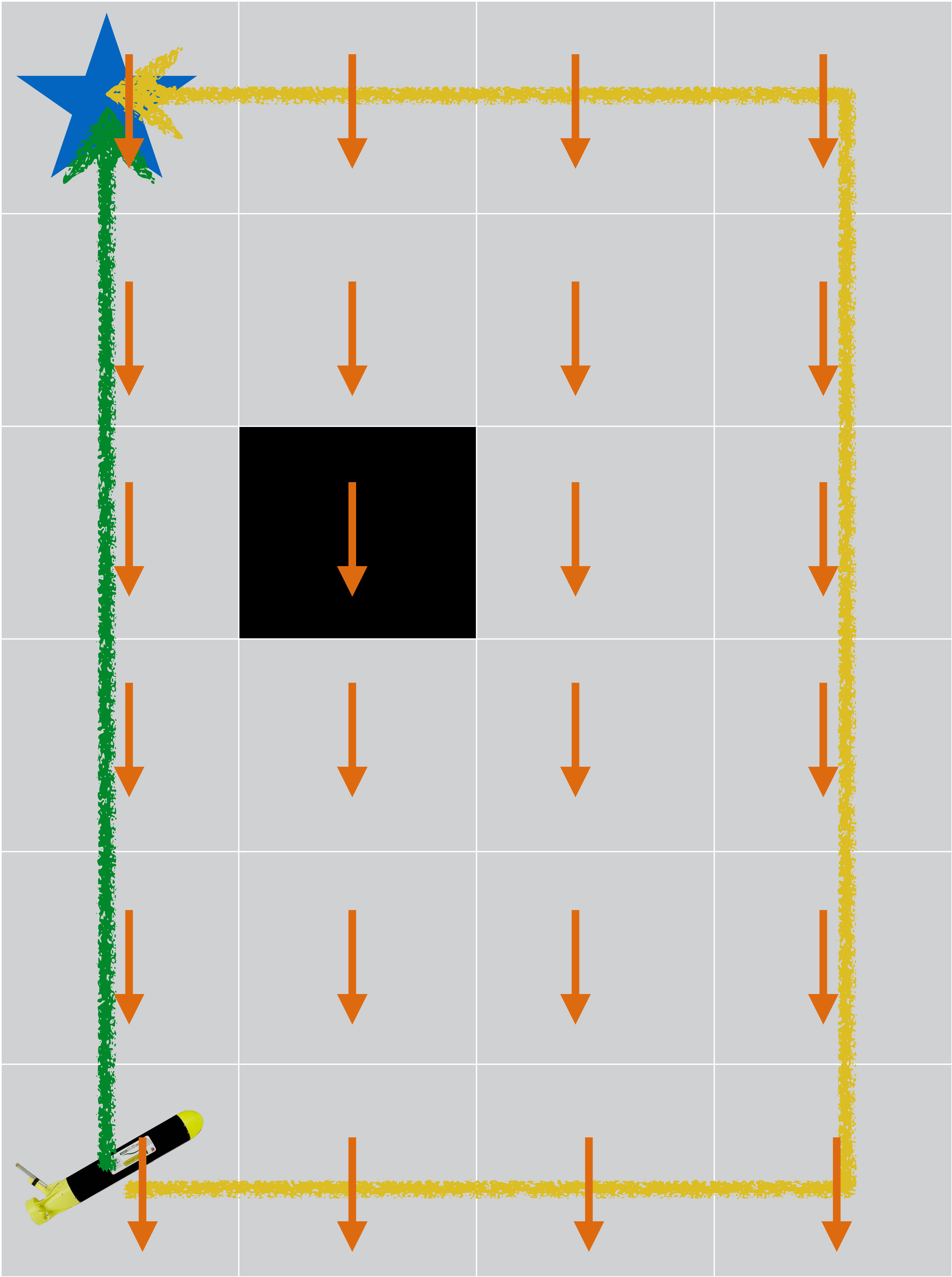


Example

South currents - P_S

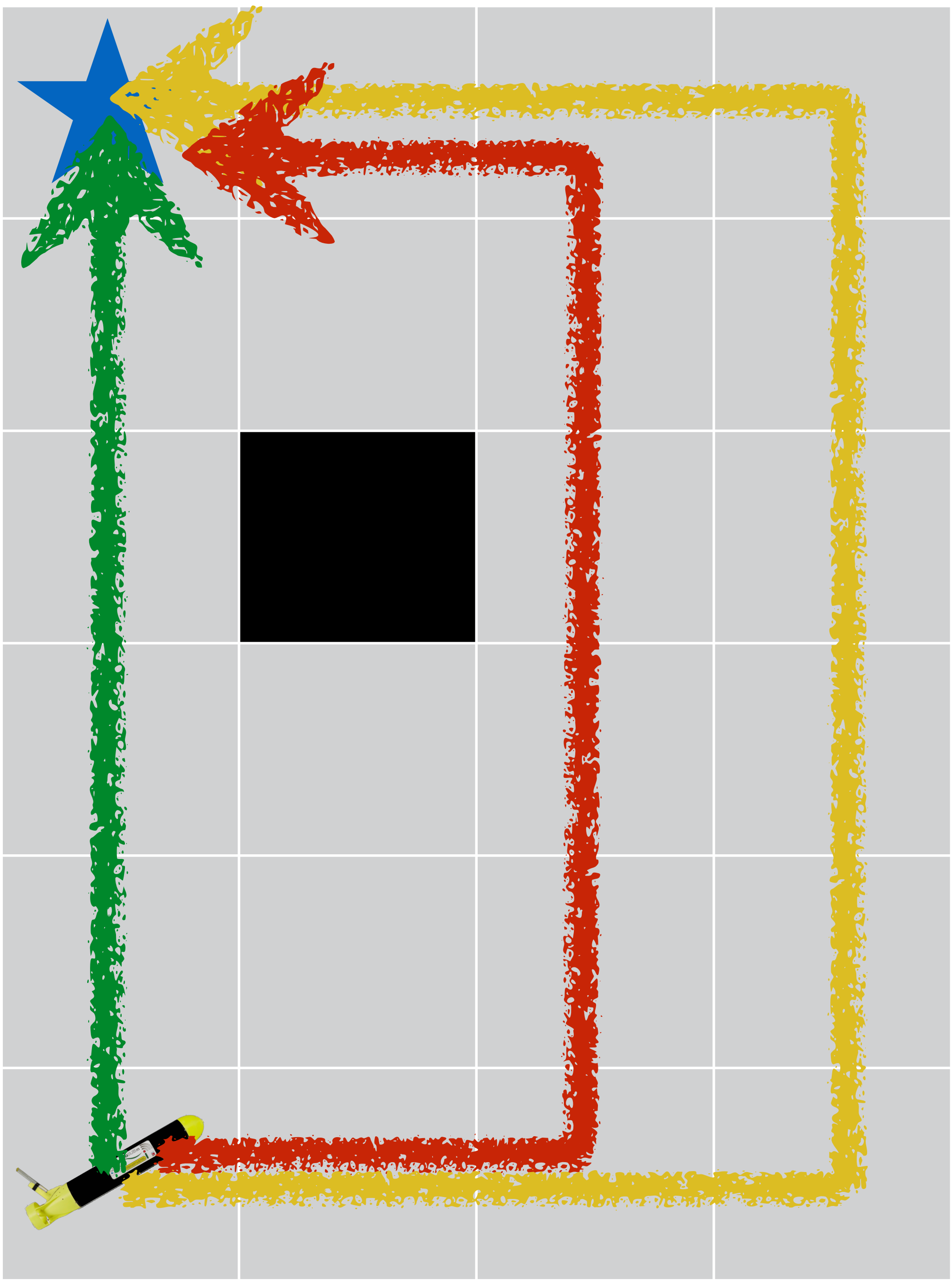
$V^{\pi_g, P_S}(s_0) = 6.25$

$V^{\pi_y, P_S}(s_0) = 13$



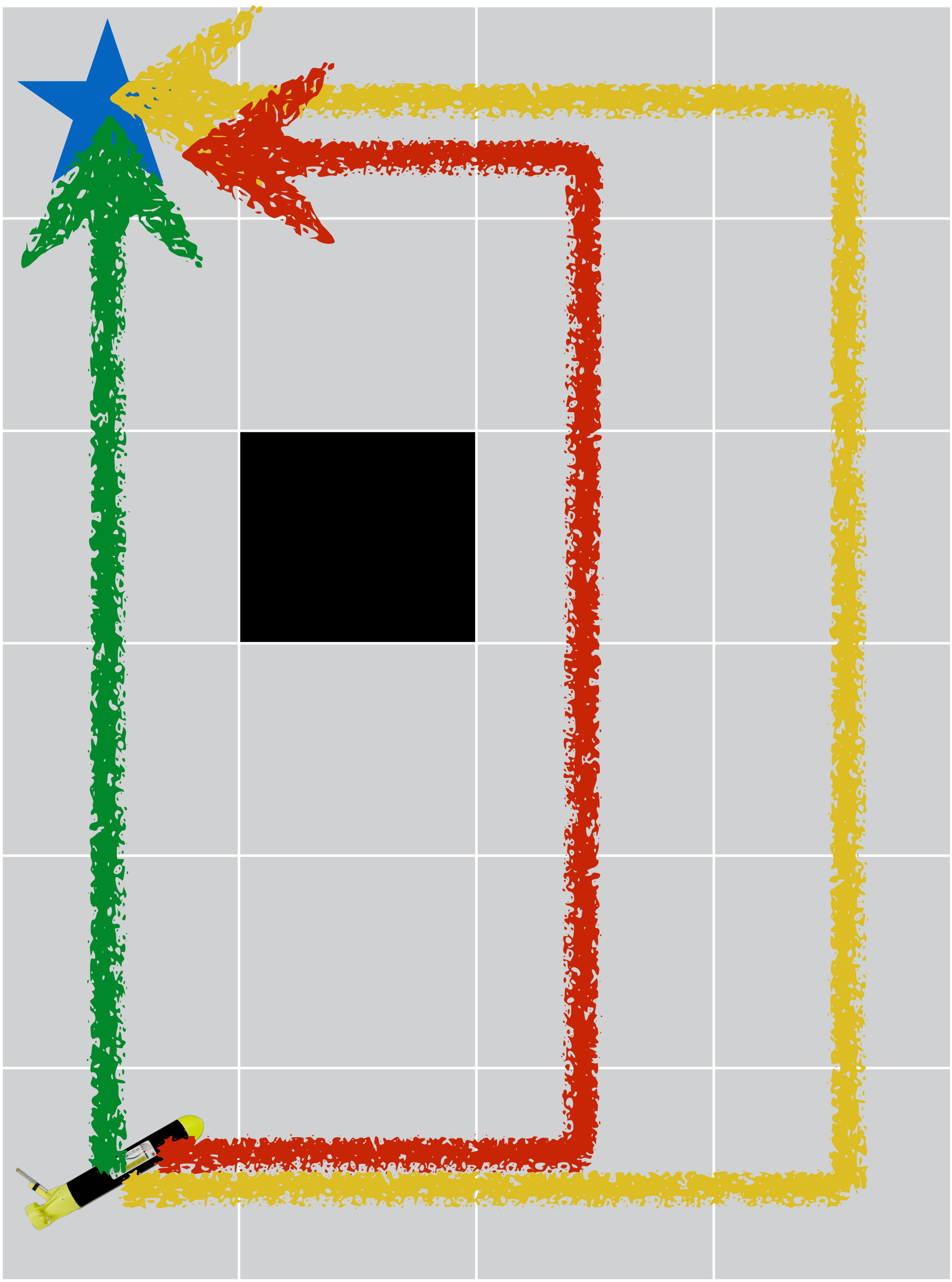
Example

$V^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S
	5	4.31	5	14.05	6.25
	11	9.4	10.89	11.05	13
	9	7.67	16.61	9.69	10.75



Example

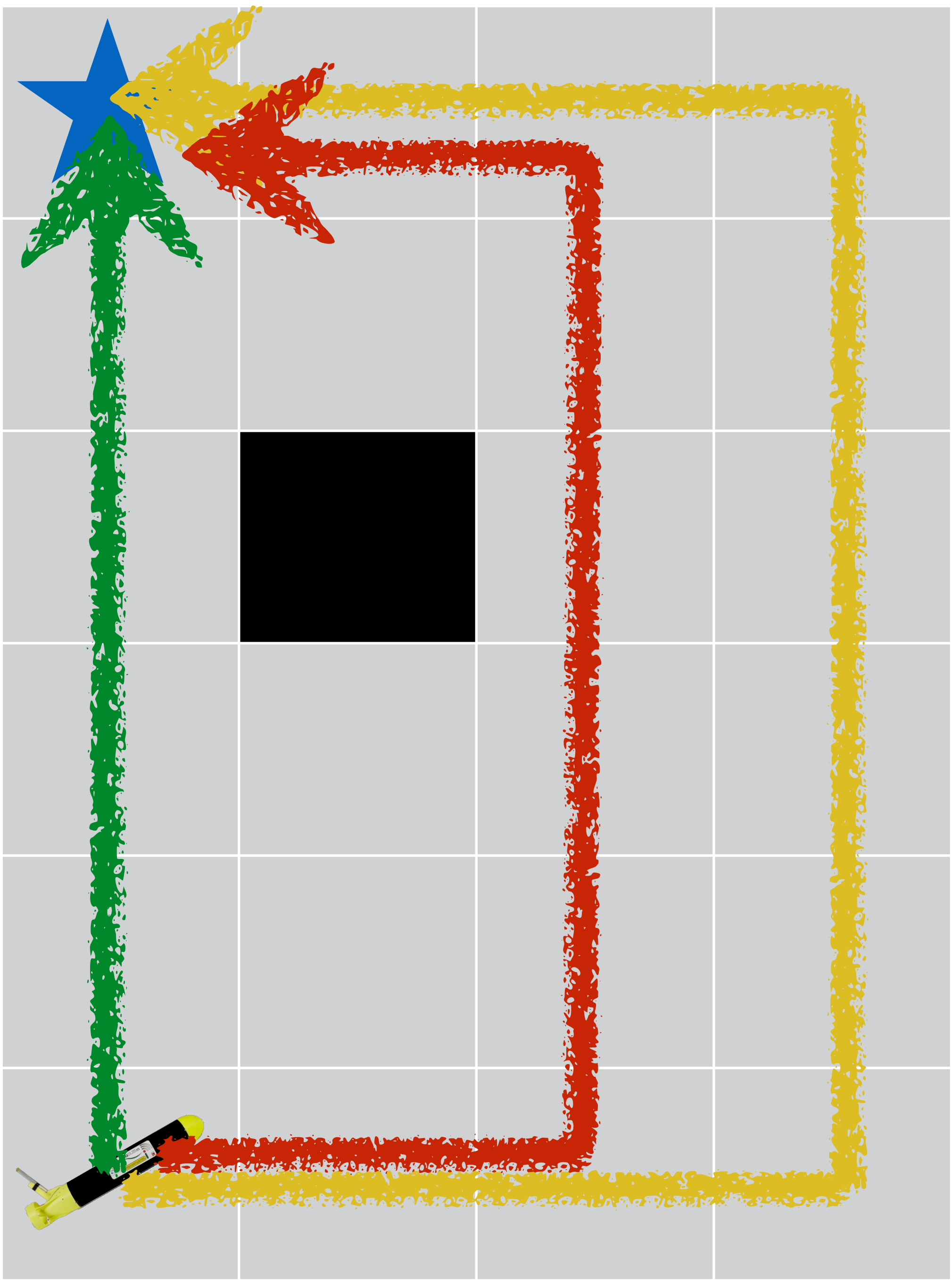
$V^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S	**Max**
	**5**	**4.31**	**5**	**14.05**	**6.25**	14.05
	**11**	**9.4**	10.89	11.05	**13**	13
	9	7.67	**16.61**	**9.69**	10.75	16.61



$$V^{wc}(s) = \min_{\pi \in \Pi} \max_{P \in \{P_1, \ldots, P_n\}} V^{\pi,P}(s)$$

Example

$V^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S	**Max**
	**5**	**4.31**	**5**	**14.05**	**6.25**	14.05
	**11**	**9.4**	10.89	11.05	**13**	13
	9	7.67	**16.61**	**9.69**	10.75	16.61



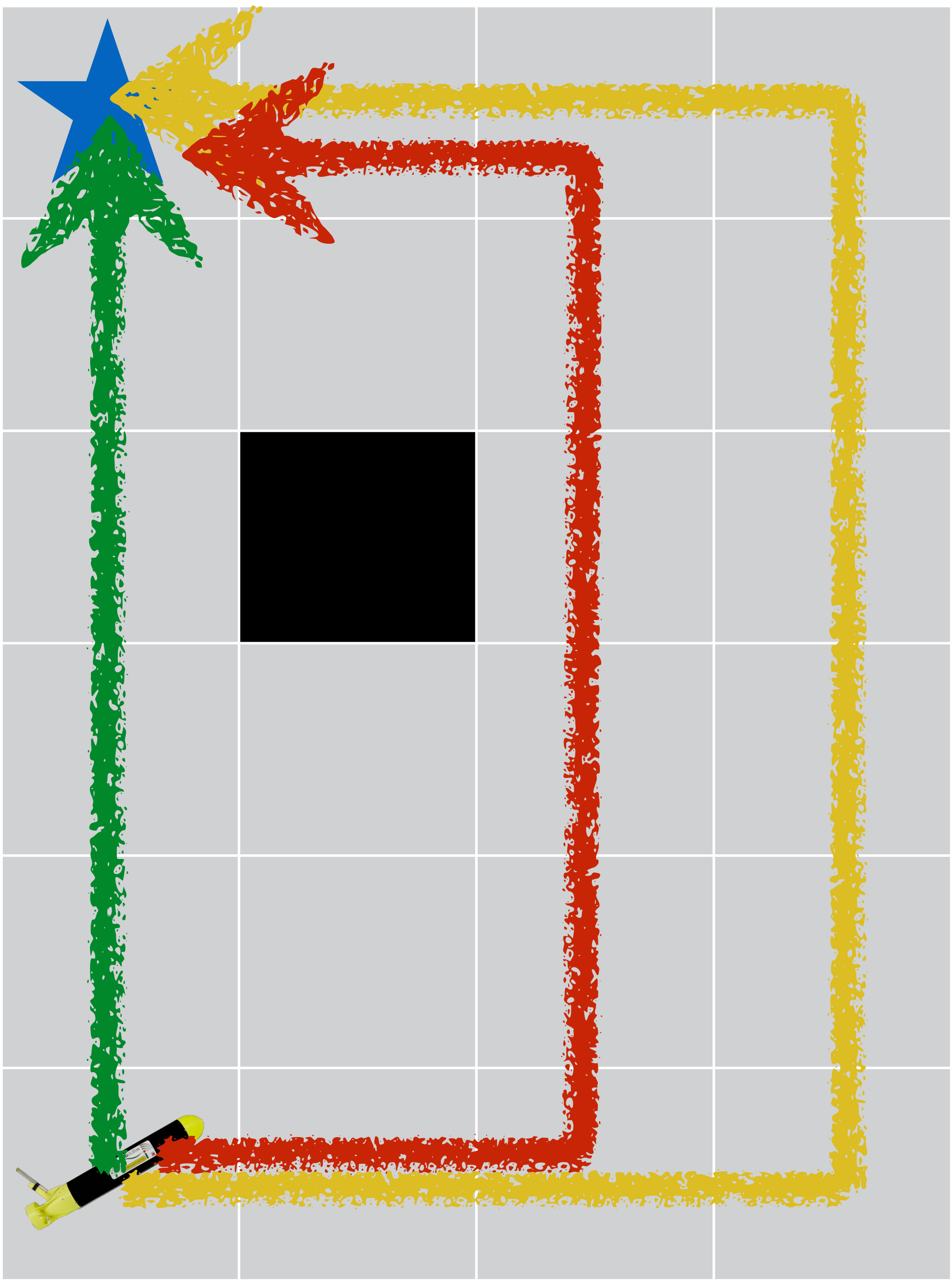
$$V^{wc}(s) = \min_{\pi \in \Pi} \max_{P \in \{P_1, \ldots, P_n\}} V^{\pi,P}(s)$$

$reg^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S
	**0**	**0**	**0**	**4.36**	**0**
	**6**	**5.09**	5.89	1.36	**6.75**
	4	3.36	**11.61**	**0**	4.5

$$reg^{\pi,P}(s) = V^{\pi,P}(s) - V^{*,P}(s)$$

Example

$V^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S	**Max**
	5	**4.31**	**5**	**14.05**	**6.25**	14.05
	11	**9.4**	10.89	11.05	**13**	13
	9	7.67	**16.61**	**9.69**	10.75	16.61



$$V^{wc}(s) = \min_{\pi \in \Pi} \max_{P \in \{P_1, \ldots, P_n\}} V^{\pi,P}(s)$$

$reg^{\pi,P}(s_0)$	P_Z	P_N	P_W	P_E	P_S	**Max**
	0	**0**	**0**	**4.36**	**0**	4.36
	6	**5.09**	5.89	1.36	**6.75**	6.75
	4	3.36	**11.61**	**0**	4.5	11.61

$$reg^{\pi,P}(s) = V^{\pi,P}(s) - V^{*,P}(s)$$

$$V^{reg}(s) = \min_{\pi \in \Pi} \max_{P \in \mathscr{P}} reg^{\pi,P}(s)$$

Optimising regret in sample-based UMDPs

- For finite-horizon MDPs, one can find approximate solutions by solving an optimisation problem
  - We will see a variant of this formulated later
  - Suboptimalty bounds for the solution can be provided [Ahmed et al.'17]
- For sample-based uncertain SSPs, we do not know of a solution method
  - Not even approximate
- We will go over an approximate solution for sample-based uncertain SSPs
  - Brings ideas from robust value iteration whilst maintaining some notion of dependency between transitions

DP for policy evaluation

- We can evaluate the regret of a policy via dynamic programming

$$reg^{\pi,P}(s) = \sum_{a \in A} \pi(s,a) \,.\, [C^P_{gap}(s,a) + \sum_{s' \in S} P(s,a,s') \,.\, reg^{\pi,P}(s')]$$

where $reg^{\pi,P}(g) = 0$ for all $g \in goal$

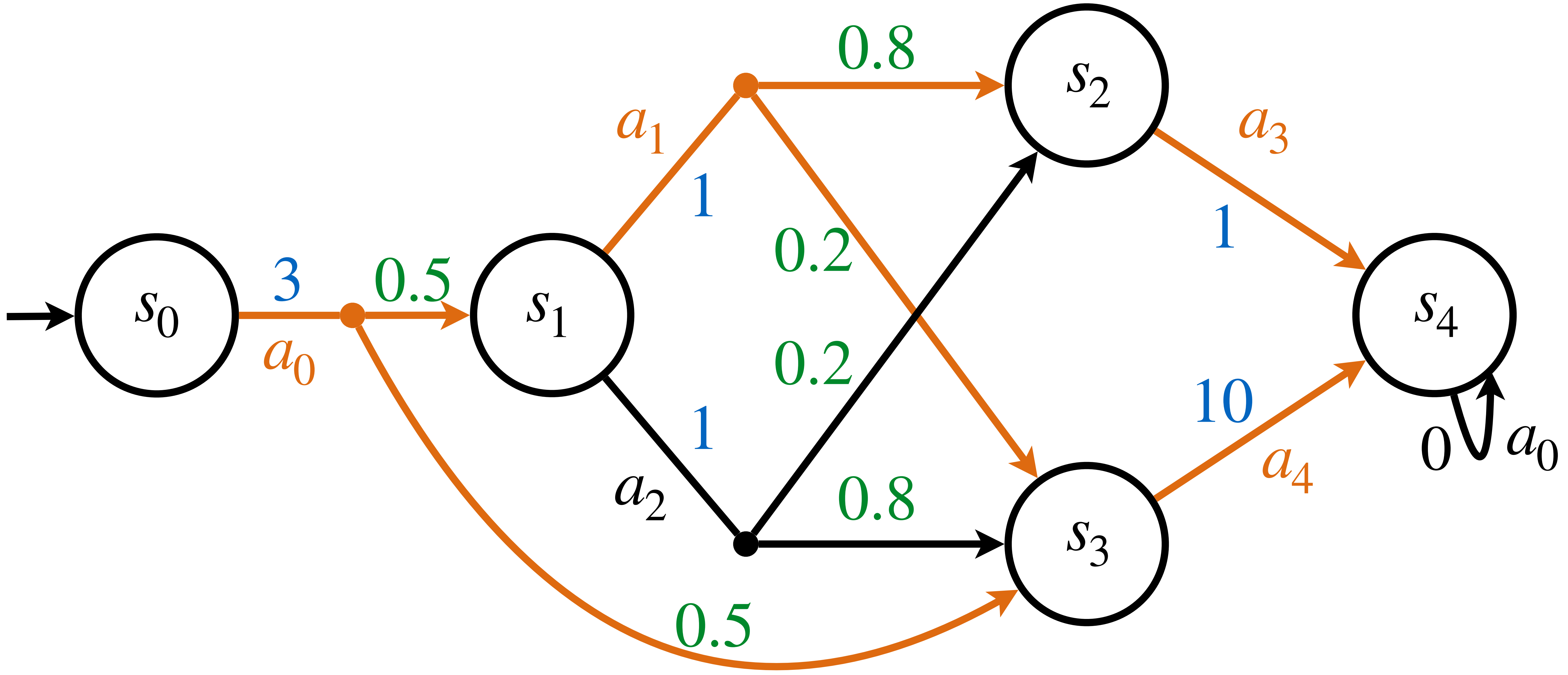
and $C^P_{gap}(s,a) = [C(s,a) + \sum_{s' \in S} P(s,a,s') \,.\, V^{*,P}(s')] - V^{*,P}(s)$

Q-value of action a, $Q(s,a)$

Optimal vale

- $C^P_{gap}(s,a)$ is the gap between
  - Taking action a at s and then following the optimal policy
  - Following the optimal policy from s

Example



$V^{*,P}(s_4) = 0$

$V^{*,P}(s_3) = 10$

$V^{*,P}(s_2) = 1$

$V^{*,P}(s_1) = 1 + 0.8V^{*,P}(s_2) + 0.2V^{*,P}(s_3) = 3.8$

$V^{*,P}(s_0) = 3 + 0.5V^{*,P}(s_1) + 0.5V^{*,P}(s_3) = 9.9$

- Regret of π, which takes action a_2 in s_1:

$$reg^{\pi,P}(s) = \sum_{a \in A} \pi(s,a) . [C^P_{gap}(s,a) + \sum_{s' \in S} P(s,a,s') . reg^{\pi,P}(s')]$$

where $reg^{\pi,P}(g) = 0$ for all $g \in goal$

and $C^P_{gap}(s,a) = [C(s,a) + \sum_{s' \in S} P(s,a,s') . V^{*,P}(s')] - V^{*,P}(s)$

$reg^{\pi,P}(s_2) = 0$

$reg^{\pi,P}(s_3) = 0$

$reg^{\pi,P}(s_4) = 0$

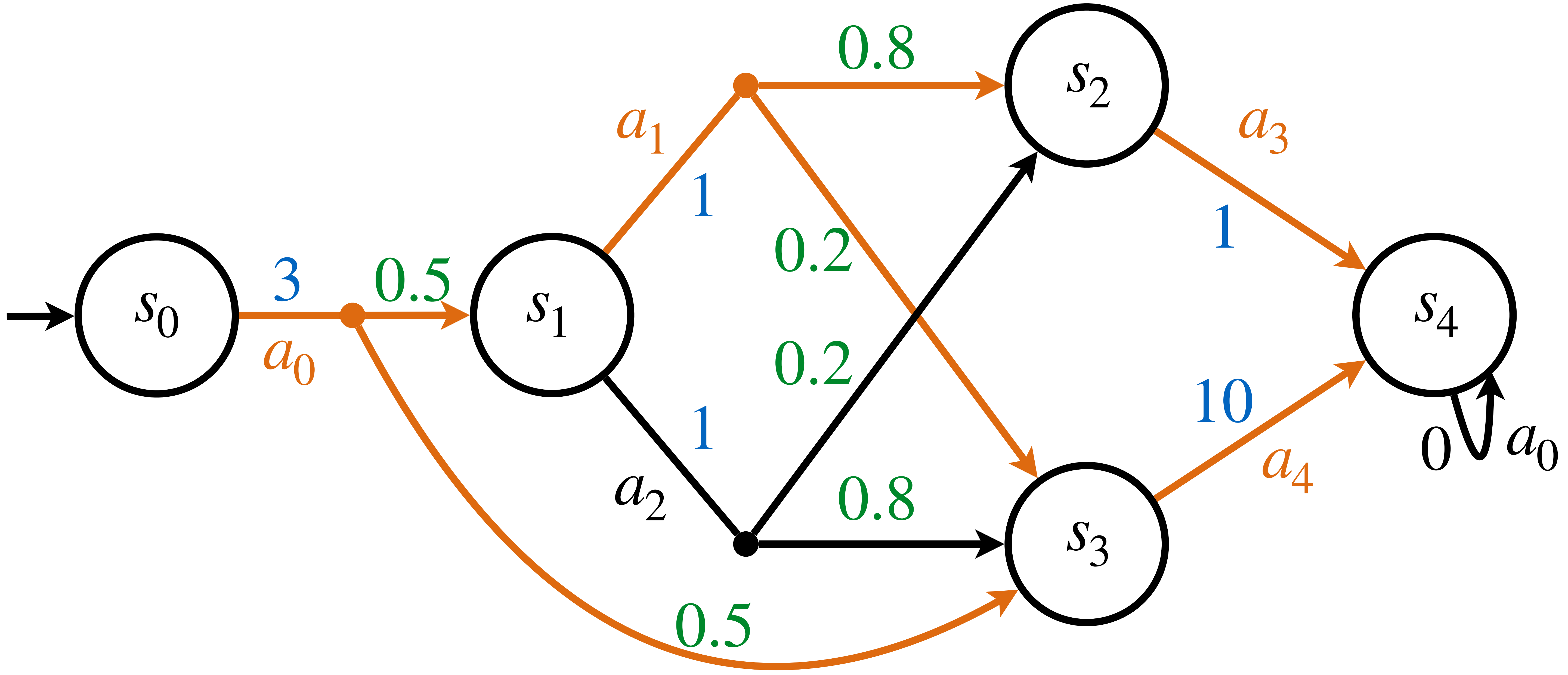
$C^P_{gap}(s_1, a_2) = [1 + 0.8 \cdot 10 + 0.2 \cdot 1] - 3.8 = 5.4$

$reg^{\pi,P}(s_1) = 5.4 + 0.8 \cdot 0 + 0.2 \cdot 0 = 5.4$

$C^P_{gap}(s_0, a_0) = [3 + 0.5 \cdot 3.8 + 0.5 \cdot 10] - 9.9 = 0$

$reg^{\pi,P}(s_0) = 0 + 0.5 \cdot 0 + 0.5 \cdot 5.4 = 2.7$

Example



$$V^{*,P}(s_4) = 0$$
$$V^{*,P}(s_3) = 10$$
$$V^{*,P}(s_2) = 1$$
$$V^{*,P}(s_1) = 1 + 0.8V^{*,P}(s_2) + 0.2V^{*,P}(s_3) = 3.8$$
$$V^{*,P}(s_0) = 3 + 0.5V^{*,P}(s_1) + 0.5V^{*,P}(s_3) = 9.9$$

- Regret of π, which takes action a_2 in s_1:

$$reg^{\pi,P}(s) = \sum_{a \in A} \pi(s,a)\,.\,[C^P_{gap}(s,a) + \sum_{s' \in S} P(s,a,s')\,.\,reg^{\pi,P}(s')]$$

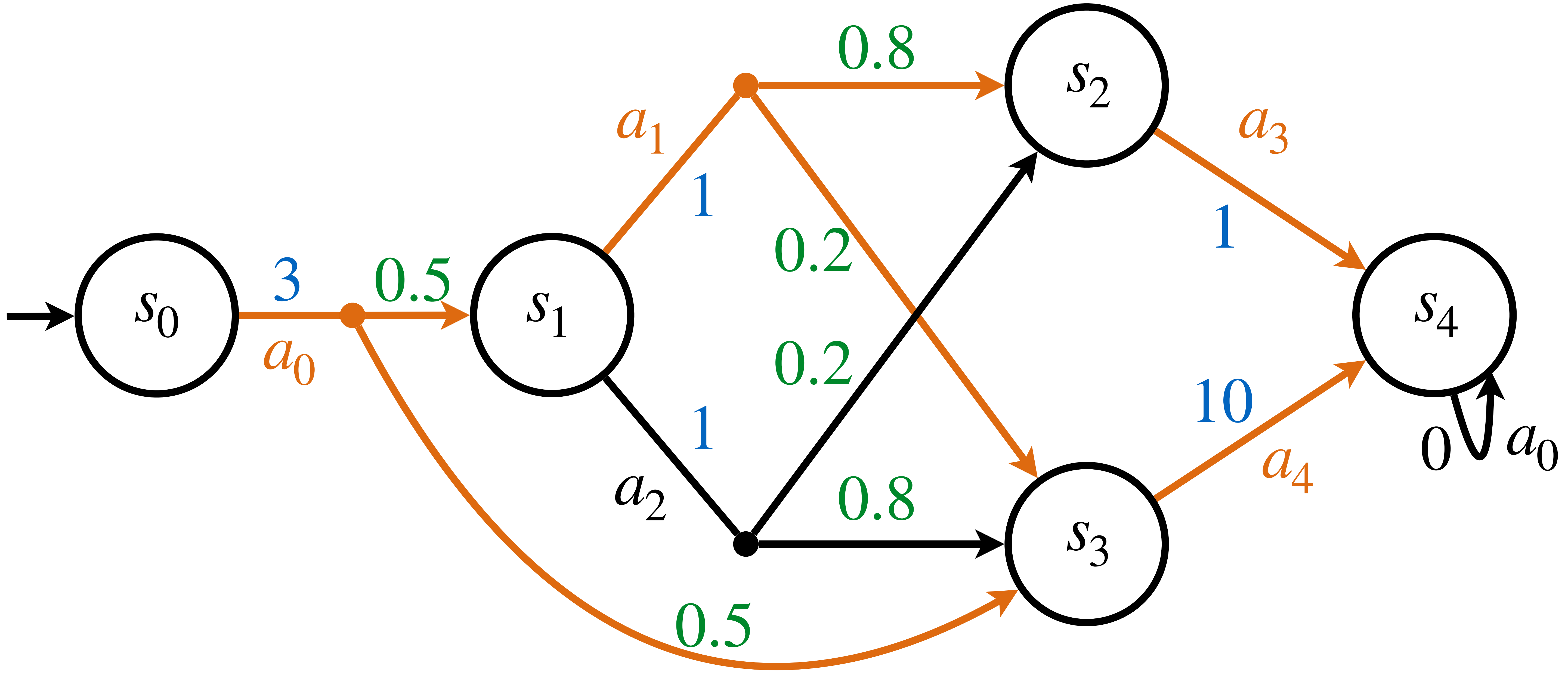
where $reg^{\pi,P}(g) = 0$ for all $g \in goal$

and $C^P_{gap}(s,a) = [C(s,a) + \sum_{s' \in S} P(s,a,s')\,.\,V^{*,P}(s')] - V^{*,P}(s)$

$$V^{\pi,P}(s_0) = 3 + 0.5 \cdot 10 + 0.5 \cdot (1 + 0.2 \cdot 1 + 0.8 \cdot 10) = 12.6$$
$$reg^{\pi,P}(s_0) = V^{\pi,P}(s_0) - V^{*,P}(s_0) = 12.6 - 9.9 = 2.7$$

$$reg^{\pi,P}(s_2) = 0$$
$$reg^{\pi,P}(s_3) = 0$$
$$reg^{\pi,P}(s_4) = 0$$
$$C^P_{gap}(s_1, a_2) = [1 + 0.8 \cdot 10 + 0.2 \cdot 1] - 3.8 = 5.4$$
$$reg^{\pi,P}(s_1) = 5.4 + 0.8 \cdot 0 + 0.2 \cdot 0 = 5.4$$
$$C^P_{gap}(s_0, a_0) = [3 + 0.5 \cdot 3.8 + 0.5 \cdot 10] - 9.9 = 0$$
$$reg^{\pi,P}(s_0) = 0 + 0.5 \cdot 0 + 0.5 \cdot 5.4 = 2.7$$

Example



$V^{*,P}(s_4) = 0$

$V^{*,P}(s_3) = 10$

$V^{*,P}(s_2) = 1$

$V^{*,P}(s_1) = 1 + 0.8V^{*,P}(s_2) + 0.2V^{*,P}(s_3) = 3.8$

$V^{*,P}(s_0) = 3 + 0.5V^{*,P}(s_1) + 0.5V^{*,P}(s_3) = 9.9$

- Regret of π, which takes action a_2 in s_1:

$$reg^{\pi,P}(s) = \sum_{a \in A} \pi(s,a) . [C^P_{gap}(s,a) + \sum_{s' \in S} P(s,a,s') . reg^{\pi,P}(s')]$$

where $reg^{\pi,P}(g) = 0$ for all $g \in goal$

and $C^P_{gap}(s,a) = [C(s,a) + \sum_{s' \in S} P(s,a,s') . V^{*,P}(s')] - V^{*,P}(s)$

$V^{\pi,P}(s_0) = 3 + 0.5 \cdot 10 + 0.5 \cdot (1 + 0.2 \cdot 1 + 0.8 \cdot 10) = 12.6$

$reg^{\pi,P}(s_0) = V^{\pi,P}(s_0) - V^{*,P}(s_0) = 12.6 - 9.9 = 2.7$

$reg^{\pi,P}(s_2) = 0$

$reg^{\pi,P}(s_3) = 0$

$reg^{\pi,P}(s_4) = 0$

$C^P_{gap}(s_1, a_2) = [1 + 0.8 \cdot 10 + 0.2 \cdot 1] - 3.8 = 5.4$

$reg^{\pi,P}(s_1) = 5.4 + 0.8 \cdot 0 + 0.2 \cdot 0 = 5.4$

$C^P_{gap}(s_0, a_0) = [3 + 0.5 \cdot 3.8 + 0.5 \cdot 10] - 9.9 = 0$

$reg^{\pi,P}(s_0) = 0 + 0.5 \cdot 0 + 0.5 \cdot 5.4 = 2.7$

Robust value iteration for regret

Compute $V^{*,P}(s)$ for all $P \in \mathcal{P}$ and $s \in S$
$\pi \leftarrow nil$
$reg^{\pi}(s) \leftarrow 0$ for all $s \in S$
repeat
  for all $s \in S$ **do**
    $reg_{old}(s) \leftarrow reg^{\pi}(s)$
    $$reg^{\pi}(s) \leftarrow \min_{a \in A} \max_{P \in \mathcal{P}} \left[C^{P}_{gap}(s,a) + \sum_{s' \in S} P(s,a,s') reg^{P,\pi}(s') \right]$$
    $$\pi(s) \leftarrow \arg\min_{a \in A} \max_{P \in \mathcal{P}} \left[C^{P}_{gap}(s,a) + \sum_{s' \in S} P(s,a,s') reg^{P,\pi}(s') \right]$$
  end for
until $\max_{s \in S}(|reg^{\pi}(s) - reg_{old}(s)|) < \epsilon$

Can be solved by checking all combinations of a and P

Regret for s' calculated under the same P as s

- Choose different P per step
  - Assumes rectangularity - too much power to the environment
  - Next, we will fix P for n steps
- Even in $\mathscr{P}$ is rectangular, solution is approximate
  - Expected, as optimising regret is hard, even for rectangular uncertainty sets

N-step options

- An n-step option is defined as $o = (s_0, \pi^o, goal^o, n)$, where:
  - $s_0 \in S$ is the initiation state
  - $\pi^o : S \rightarrow A$ is the option policy
  - $goal^o \subseteq S$ is a set of termination states
  - $n \in \mathbb{N}$ is the maximum number of steps

- An n-step option is executed until
  - A state in $goal^o$ is reached, or
  - n steps have occurred

- Analysing the Markov chain induced by applying o for n steps, we can compute:
  - The state distribution after applying o in $s \in \bar{s}$ for n steps, $Pr(s' \mid s, o)$
  - The expected cumulative cost for applying o in $s \in \bar{s}$ for n steps, $V_n^{o,P}(s)$

N-step option MDP

- For $\mathscr{M} = (S, s_0, A, P, C, goal)$ with $P \in \mathscr{P}$, we define the n-step option MDP as $\mathscr{M}_n^o = (S, O_n, P^o, C_{gap}^{o,P}, goal)$, where:
  - O_n is the set of all n-step options starting in some $s \in S$ and goal termination condition $goal^o = goal$
  - $P^o : S \times O_n \rightarrow Distr(S)$ is the transition function such that for $o = (\bar{s}, \pi^o, G^o, n)$:

$$P^o(s, o, s') = \begin{cases} Pr(s' \mid s, o) & \text{if } s = \bar{s} \\ 0 & \text{otherwise} \end{cases}$$

  - $$C_{gap}^{o,P}(s, o) = \left[V_n^{o,P}(s) + \sum_{s' \in S} P^o(s, o, s') \,.\, V^{*,P}(s') \right] - V^{*,P}(s)$$

- Policy for n-step option MDP $\sigma : S \rightarrow O_n$ maps state to option to be applied

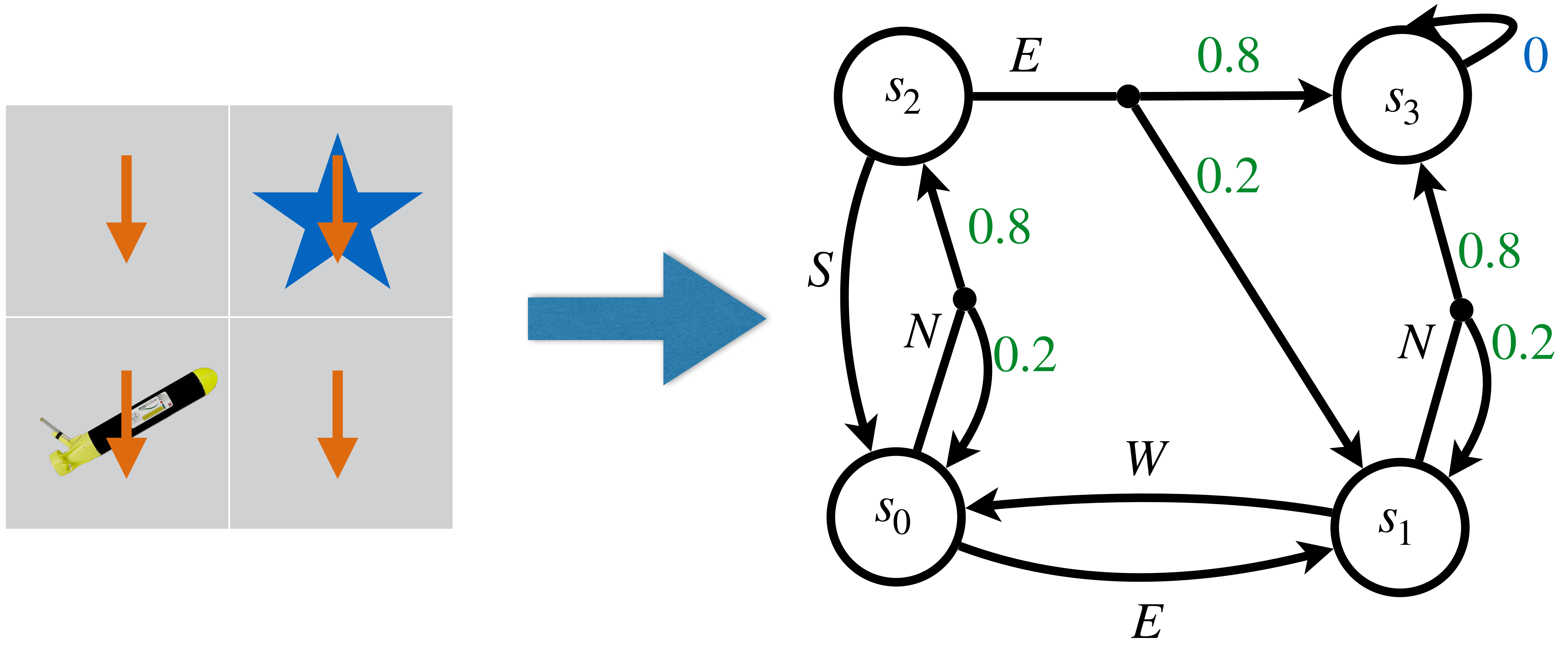
Example

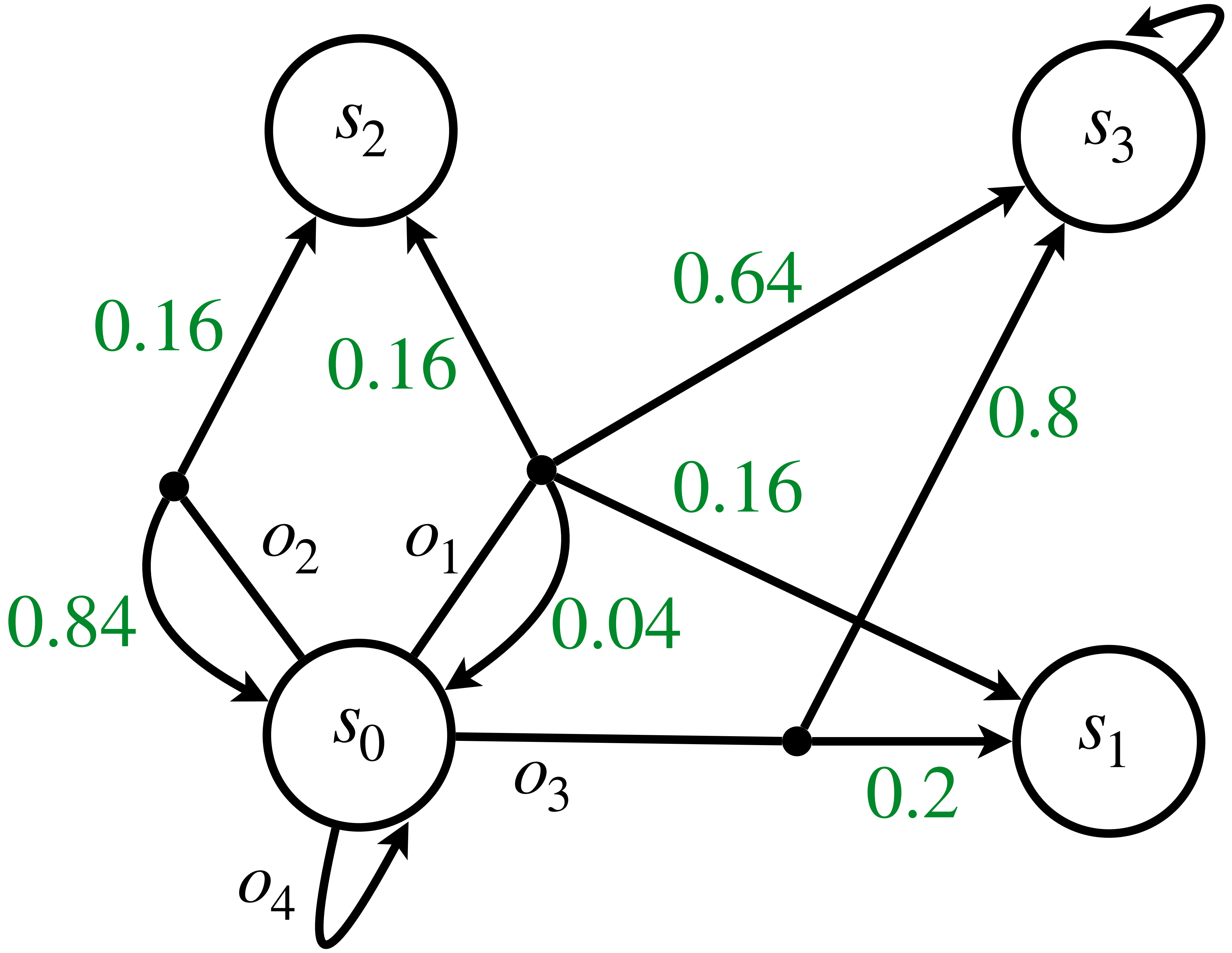
$$o_1 : s_0 \mapsto N$$
$$s_2 \mapsto E$$

$$o_2 : s_0 \mapsto N$$
$$s_2 \mapsto S$$

$$o_3 : s_0 \mapsto E$$
$$s_1 \mapsto N$$

$$o_4 : s_0 \mapsto E$$
$$s_1 \mapsto W$$





N-step option MDP

$$reg^{\pi,P}(s) = C^{P}_{gap}(s, \pi(s)) + \sum_{s' \in S} P(s, \pi(s), s') \,.\, reg^{\pi,P}(s')]$$

$$reg^{\sigma,P^o}(s) = C^{o,P}_{gap}(s, \sigma(s)) + \sum_{s' \in S} P^o(s, \sigma(s), s') \,.\, reg^{\sigma,P^o}(s')]$$

N-step dependency robust value iteration for regret

Compute $V^{*,P}(s)$ for all $P \in \mathcal{P}$ and $s \in S$
$\sigma \leftarrow nil$
$reg^{\sigma}(s) \leftarrow 0$ for all $s \in S$
repeat
  $e \leftarrow 0$
  for all $s \in S$ **do**
    $reg_{old} \leftarrow reg^{\sigma}(s)$
    $$reg^{\sigma}(s) \leftarrow \min_{o \in O_n} \max_{P \in \mathcal{P}} \left[C^{o,P}_{gap}(s,o) + \sum_{s' \in S} P^{o}(s,o,s') reg^{\sigma}(s') \right]$$
    $$\sigma(s) \leftarrow \arg\min_{o \in O_n} \max_{P \in \mathcal{P}} \left[C^{o,P}_{gap}(s,o) + \sum_{s' \in S} P^{o}(s,o,s') reg^{\sigma}(s') \right]$$
    $e \leftarrow \max(e, |reg^{\sigma}(s) - reg_{old}|)$
  end for
until $e < \epsilon$

Can be solved by checking all combinations of o and P^{o}. However, enumerating all n-step options is too expensive

- Inner problem is a finite-horizon regret optimisation
  - We will pose as optimisation problem

N-step regret as optimisation

$$\min \quad reg^{\sigma}(s_{cur})$$

$$s.t. \quad reg^{\sigma}(s_{cur}) \geq V_n^P(s_{cur},0) + c^P(s_{cur},0) - V^{*,P}(s_{cur}) \qquad \forall P \in \mathscr{P}$$

$$Q_n^P(s,a,t) = C(s,a) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t = n-1$$

$$Q_n^P(s,a,t) = C(s,a) + \sum_{s' \in S} P(s,a,s') Q_n^P(s,a,t+1) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t < n-1$$

$$V_n^P(s,t) = \sum_{a \in A} \pi^o(s,a,t) \cdot Q_n^P(s,a,t) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t \leq n-1$$

$$c^P(s,a,t) = \sum_{s' \in S} P(s,a,s') \cdot c^P(s',t+1) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t < n-1$$

$$c^P(s,a,t) = \sum_{s' \in S} P(s,a,s') \cdot [reg^{\sigma}(s') + V^{*,P}(s')] \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t = n-1$$

$$c^P(s,t) = \sum_{a \in A} \pi^o(s,a,t) \cdot c^P(s,a,t) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t \leq n-1$$

- Optimisation variables:
  - $reg^{\sigma}(s_{cur})$ is the total regret we wish to minimise
  - $\pi^o(s,a,t)$ is the randomised option policy to be applied for n steps
  - $V_n^P(s,t)$ is the value of applying π^o for n steps
  - $Q_n^P(s,a,t)$ is the value of applying a from timestep t to timestep n
  - $c^P(s,t)$ is the regret accumulated by σ after π^o has been executed, weighed by the state distribution after executing π^o
  - $c^P(s,a,t)$ backpropagates the cost from timstep n to timestep 0

N-step regret as optimisation

> Quadratic constraints. Can be approximately linearised using separable programming

$$\min \quad reg^{\sigma}(s_{cur})$$

$$s.t. \quad reg^{\sigma}(s_{cur}) \geq V_n^P(s_{cur}, 0) + c^P(s_{cur}, 0) - V^{*,P}(s_{cur}) \qquad \forall P \in \mathscr{P}$$

$$Q_n^P(s,a,t) = C(s,a) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t = n-1$$

$$Q_n^P(s,a,t) = C(s,a) + \sum_{s' \in S} P(s,a,s') Q_n^P(s,a,t+1) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t < n-1$$

$$V_n^P(s,t) = \sum_{a \in A} \pi^o(s,a,t) \cdot Q_n^P(s,a,t) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t \leq n-1$$

$$c^P(s,a,t) = \sum_{s' \in S} P(s,a,s') \cdot c^P(s', t+1) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t < n-1$$

$$c^P(s,a,t) = \sum_{s' \in S} P(s,a,s') \cdot [reg^{\sigma}(s') + V^{*,P}(s')] \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t = n-1$$

$$c^P(s,t) = \sum_{a \in A} \pi^o(s,a,t) \cdot c^P(s,a,t) \qquad \forall s \in S, a \in A, P \in \mathscr{P}, t \leq n-1$$

- Optimisation variables:
  - $reg^{\sigma}(s_{cur})$ is the total regret we wish to minimise
  - $\pi^o(s,a,t)$ is the randomised option policy to be applied for n steps
  - $V_n^P(s,t)$ is the value of applying π^o for n steps
  - $Q_n^P(s,a,t)$ is the value of applying a from timestep t to timestep n
  - $c^P(s,t)$ is the regret accumulated by σ after π^o has been executed, weighed by the state distribution after executing π^o
  - $c^P(s,a,t)$ backpropagates the cost from timstep n to timestep 0

Summary

- For many practical problems, one wants to consider dependencies between transitions
  - Breaks rectangularity assumption
  - Enables less conservative behaviour
  - Enables adaptive behaviour
  - Problem becomes hard to solve optimally
    - We looked at approximation techniques
- Regret is a suitable measure which trades-off robustness and conservatism
- We optimise for regret where we assume n-step rectangularity rather than (1-step) rectangularity

References

- Regret optimisation in sample-based UMDPs

  - A. Ahmed, P. Varakantham, M. Lowalekar, Y. Adulyasak and P. Jaillet, *Sampling Based Approaches for Minimizing Regret in Uncertain Markov Decision Processes,* JAIR, 2017.
  - M. Rigter, B. Lacerda, and N. Hawes. *Minimax Regret Optimisation for Robust Planning in Uncertain Markov Decision Processes*, AAAI, 2021.

- Related models

  - S. Mannor, O. Mebel and H. Xu, *Lightning Does Not Strike Twice: Robust MDPs with Coupled Uncertainty, ICML,* 2012.
  - L. N. Steimle, D. L. Kaufman and B. T. Denton, *Multi-model Markov decision processes*, IISE Transactions 2021.
  - M. Cubuktepe, N. Jansen, S. Junges, J.-P. Katoen and U. Topcu, *Scenario-Based Verification of Uncertain MDPs*, TACAS, 2020.