Reinforcement Learning Fundamentals

Lecture 13: Bellman Optimality

Dr Sandeep Manjanna
Assistant Professor, Plaksha University
sandeep.manjanna@plaksha.edu.in

Announcements

- There will be 4 quizzes in total and best 3 will be considered for grade calculation. **NO make-up quiz for any reason**.
- The Midterm project presentation will be online.
- There will be an in-class exam on the **15th of March.**

In today's class…

Until now…

- State and Action Value Functions
- Derived Bellman Equation

- Bellman Optimality: Optimal Value function and Optimal Policy
- Policy Extraction
- Value Iteration

Bellman Equation

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) \Big[r + \gamma v_\pi(s') \Big], \quad \text{for all } s \in \mathcal{S},
\end{aligned}
$$


Richard E. Bellman

- The Bellman equation averages over all the possibilities, weighting each by its probability of occurring.
- Linear equation in $|S|$ variables.
- Unique solution exists.

Optimal Policy

- Given:

a	b	c	d	e
10				1

  - Actions: East, West, and Exit (only available in exit states a, e)
  - Transitions: deterministic

- Quiz 1: For $\gamma = 1$, what is the optimal policy?

10				1

- Quiz 2: For $\gamma = 0.1$, what is the optimal policy?

10				1

- Quiz 3: For which γ are West and East equally good when in state d?

Optimal Policy

Define a partial ordering over policies

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \forall s$$

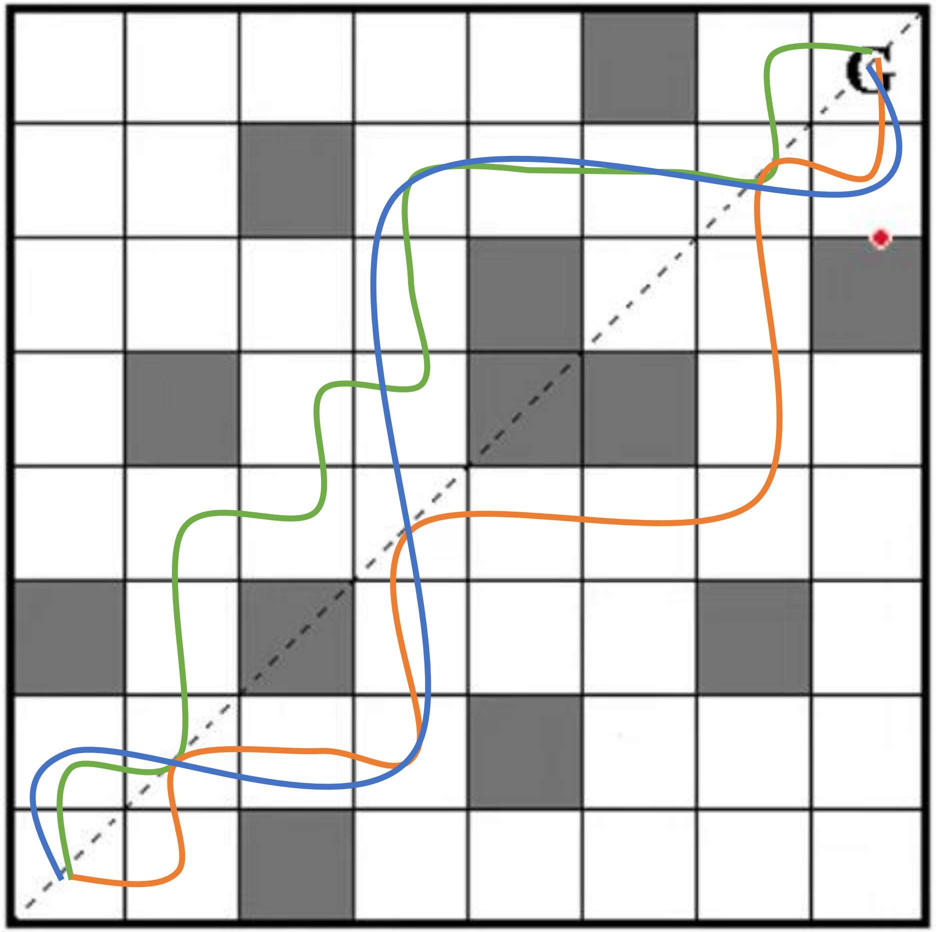
Theorem

For any Markov Decision Process

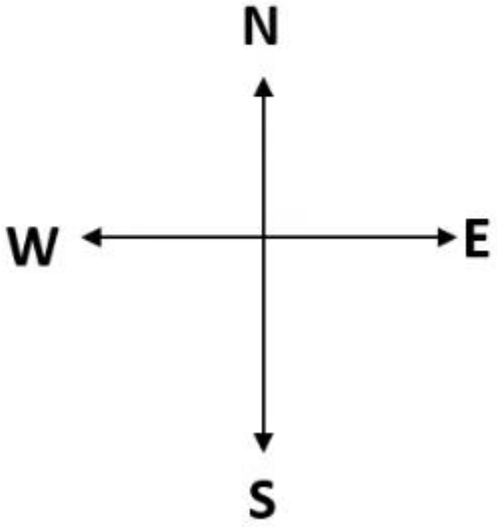
- *There exists an optimal policy* π_* *that is better than or equal to all other policies,* $\pi_* \geq \pi, \forall \pi$
- *All optimal policies achieve the optimal value function,* $v_{\pi_*}(s) = v_*(s)$
- *All optimal policies achieve the optimal action-value function,* $q_{\pi_*}(s, a) = q_*(s, a)$

Slide from Dr David Silver's course: "Reinforcement Learning."

Optimality Example



$$M = \langle S, A, p, r \rangle$$



Many optimal policies, but only one optimal value function!

Optimal Value Function

Definition

The *optimal state-value function* $v_*(s)$ is the maximum value function over all policies

$$v_*(s) = \max_{\pi} v_{\pi}(s) \quad \text{for all } s \in \mathcal{S}$$

The *optimal action-value function* $q_*(s, a)$ is the maximum action-value function over all policies

$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a) \quad \text{for all } s \in \mathcal{S} \text{ and } a \in \mathcal{A}$$

- The optimal value function specifies the best possible performance in the MDP.
- An MDP is "solved" when we know the optimal value fn.

Slide from Dr David Silver's course: "Reinforcement Learning."

Optimal Policy

An optimal policy can be found by maximising over $q_*(s, a)$,

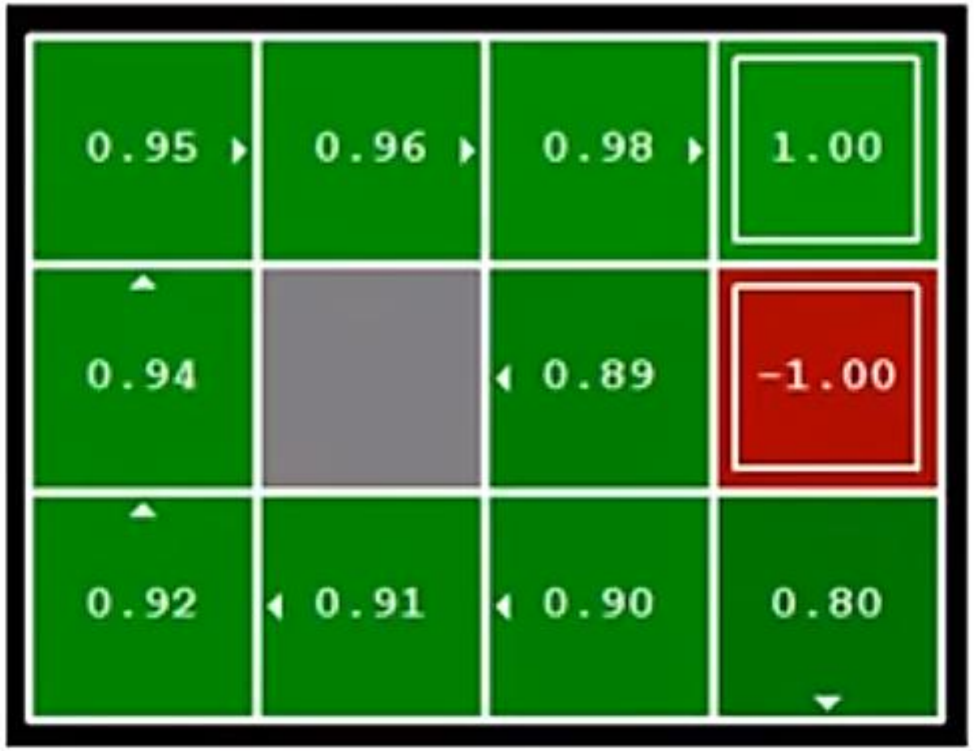
$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s, a) \\ 0 & \textit{otherwise} \end{cases}$$

- There is always a deterministic optimal policy for any MDP
- If we know $q_*(s, a)$, we immediately have the optimal policy

- Optimal value function is unique for an MDP.
- Hence, many solution approaches try to find an optimal value function, instead of directly finding the optimal policy.

Why are these useful?

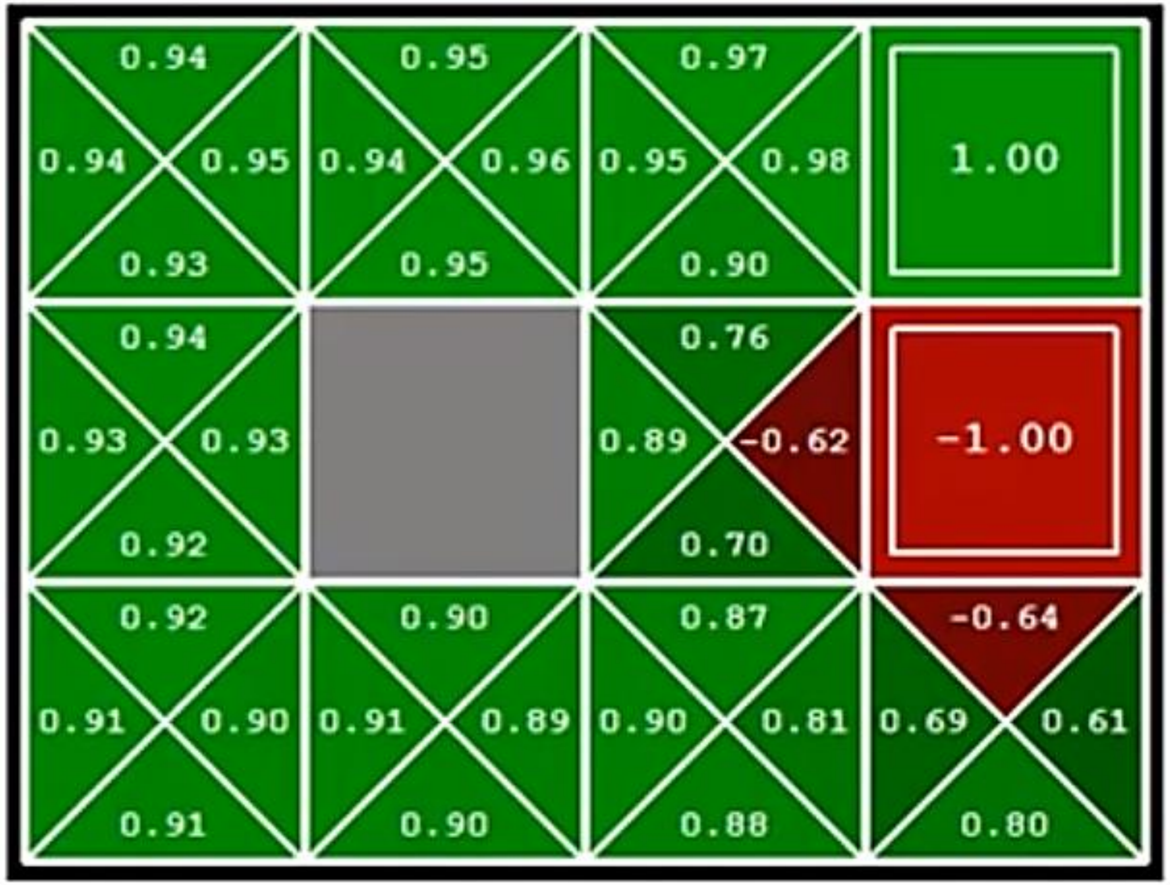
Given the optimal values v_*, how to get the optimal policy?



- We can use one-step-lookahead search to get the long-term optimal action.
- This is called **policy extraction**, since it gets the policy implied by the value.

Why are these useful?

Given the optimal q-values q_*,



- Don’t even need to do one-step-lookahead search.

Bellman Optimality Equation for v_*

$$v_\pi(s) = \sum_a \pi(a|s) \underbrace{\sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big]}_{q_\pi(s,a)}, \quad \text{for all } s \in \mathcal{S},$$

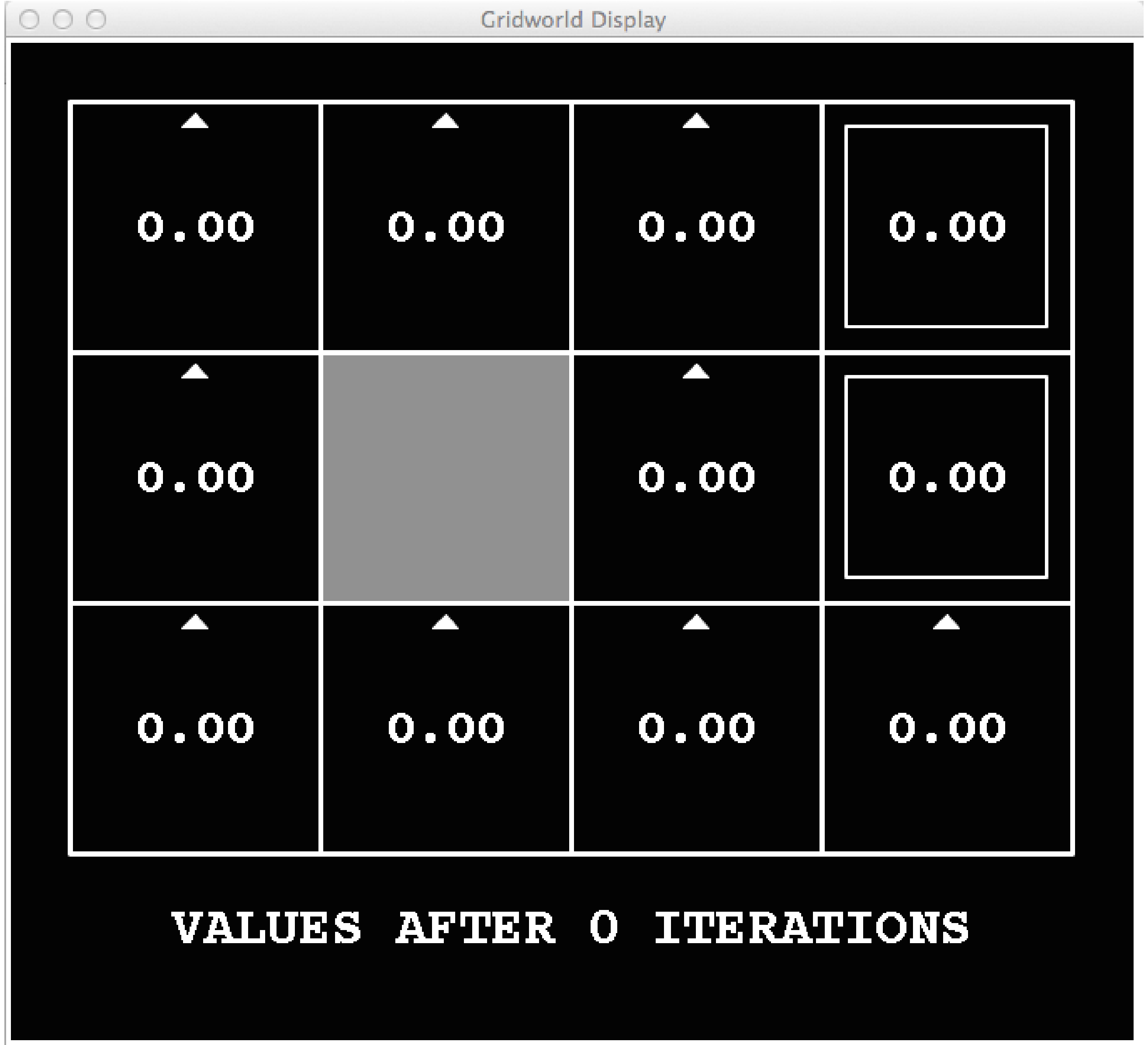
The value of a state under an optimal policy must equal the expected return for the best action from that state:

$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big]. \end{aligned}$$

Similarly,

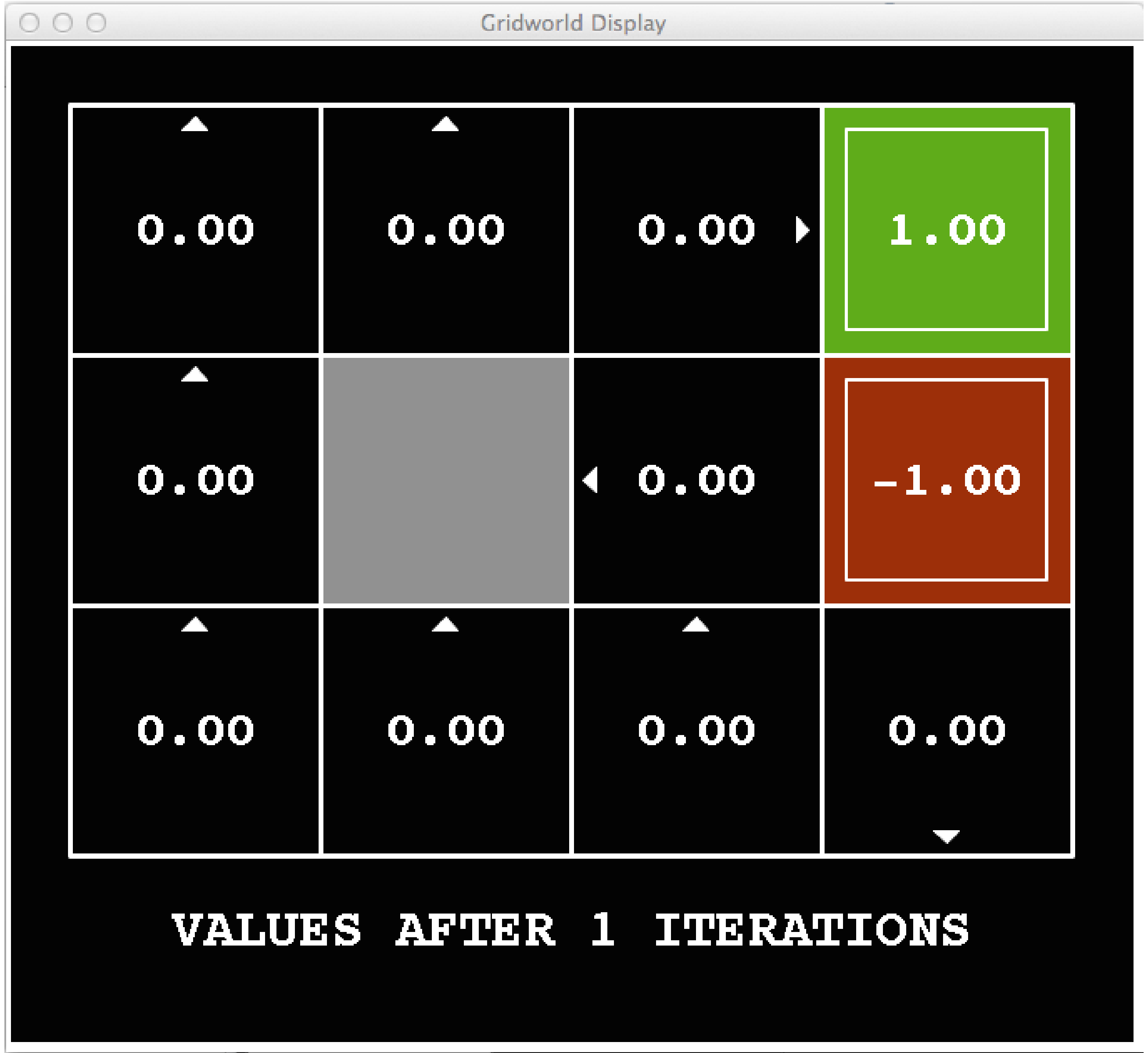
$$\begin{aligned} q_*(s,a) &= \mathbb{E}\Big[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1},a') \,\Big|\, S_t = s, A_t = a\Big] \\ &= \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \max_{a'} q_*(s',a')\Big]. \end{aligned}$$

Value Iteration



Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



Noise = 0.2
Discount = 0.9
Living reward = 0

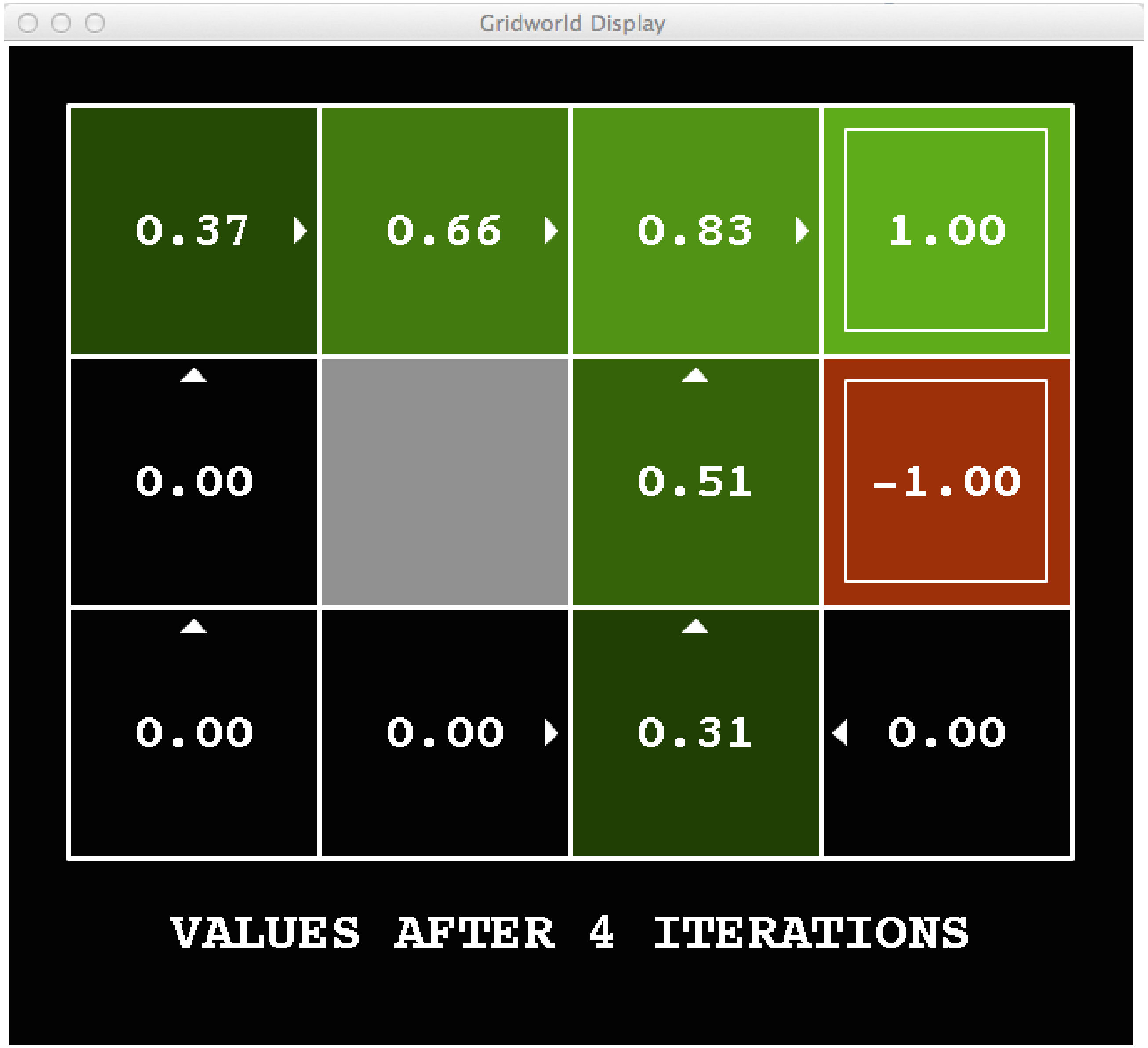
Value Iteration

Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration

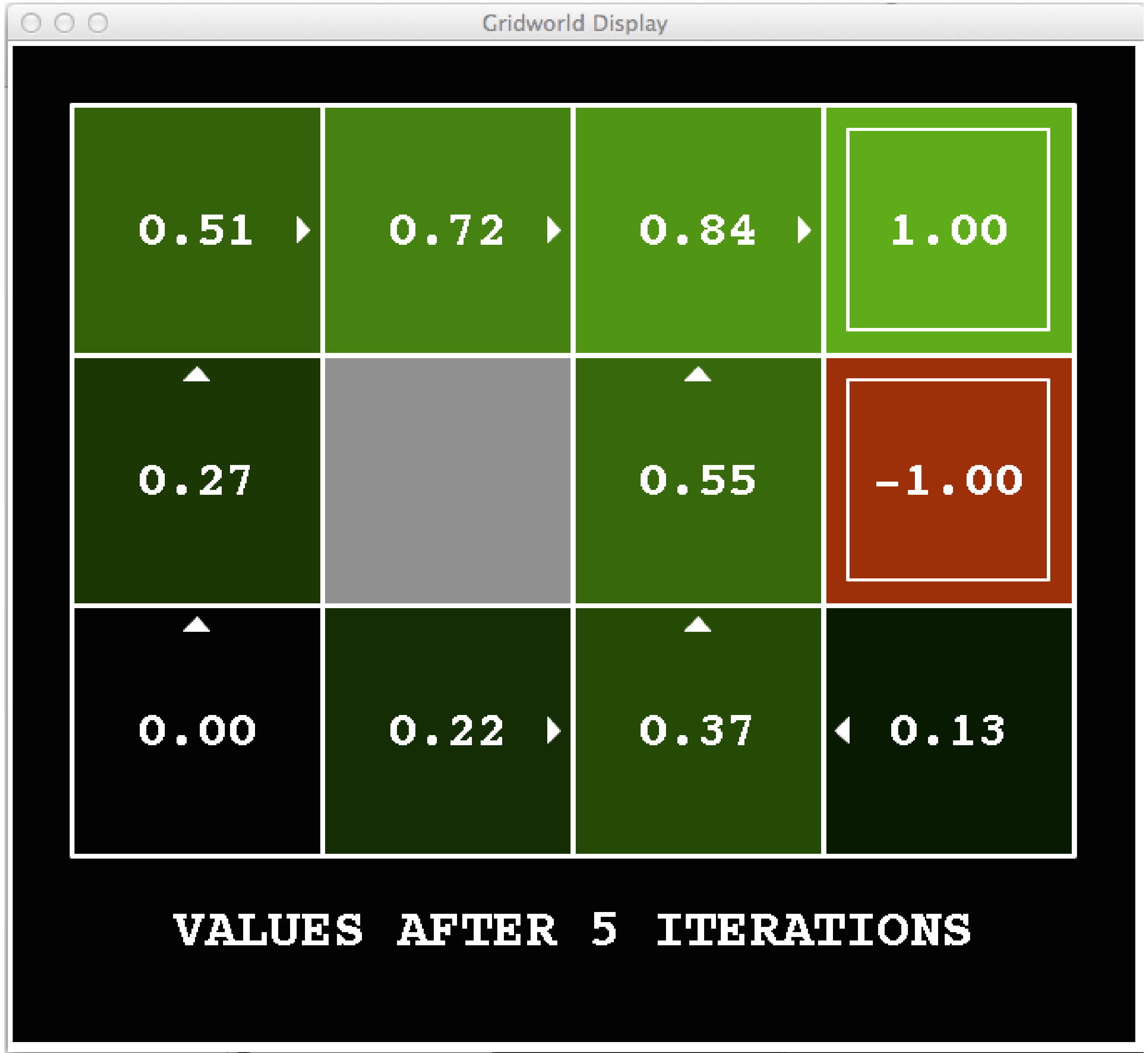
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



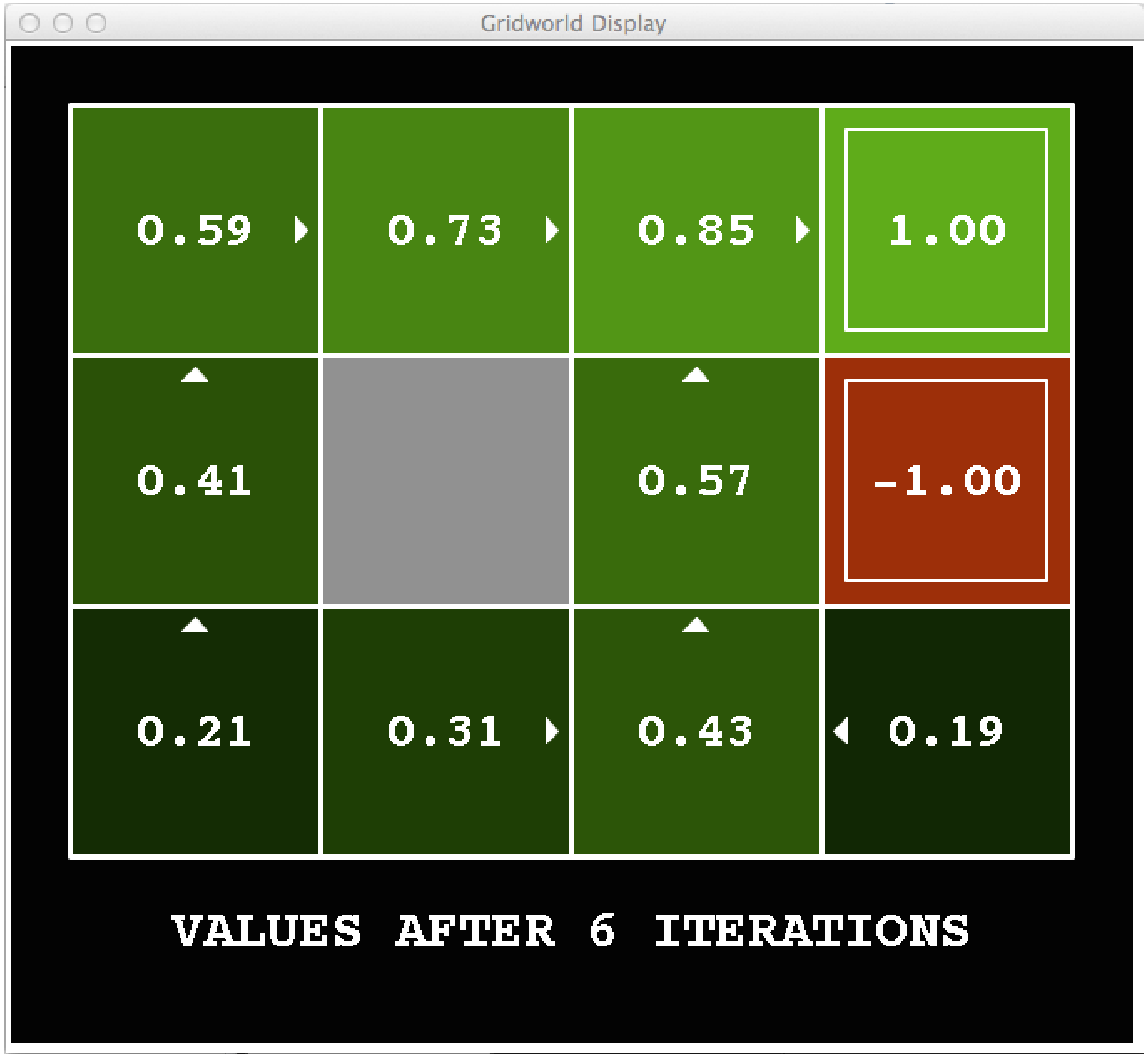
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



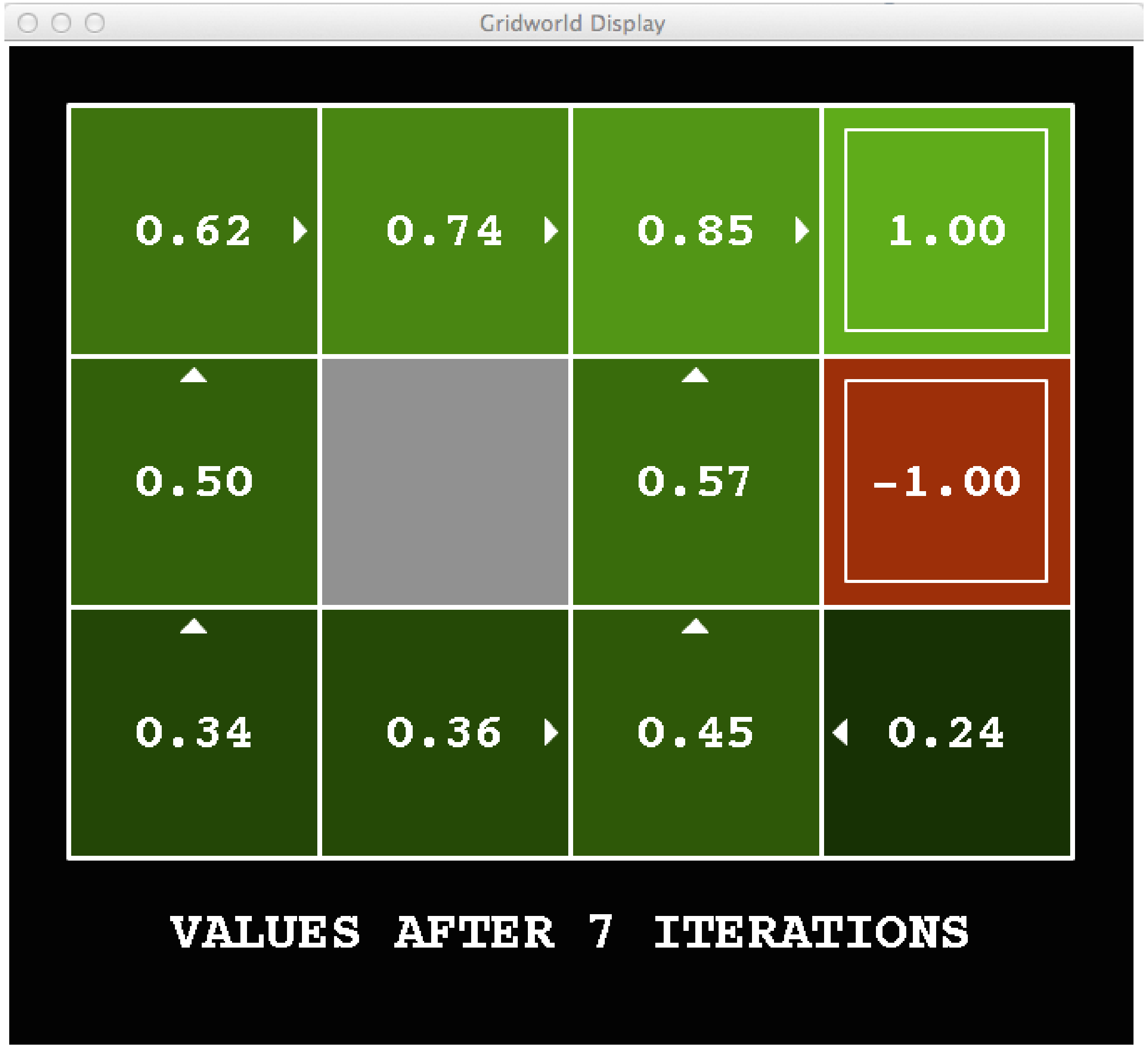
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



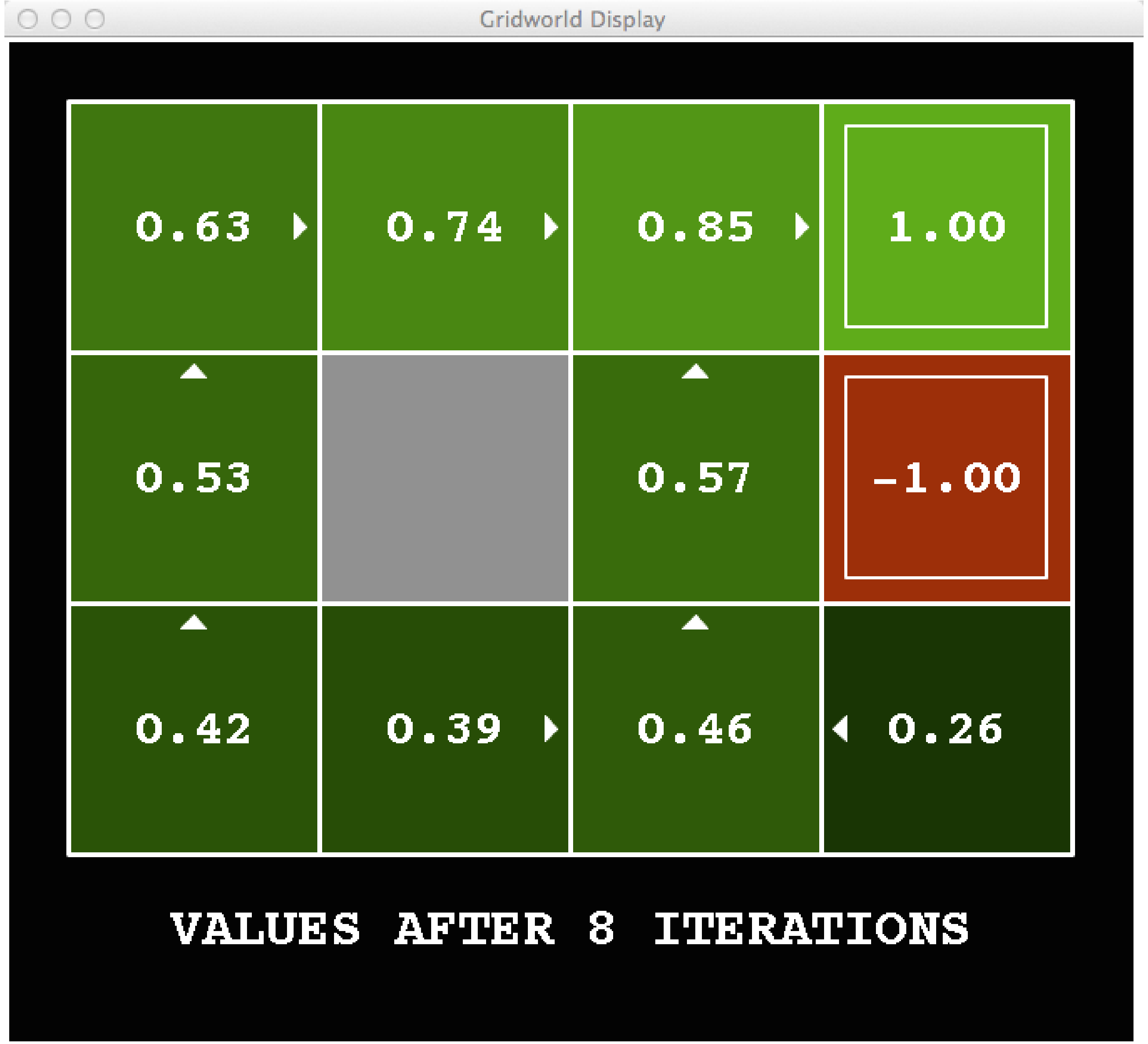
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



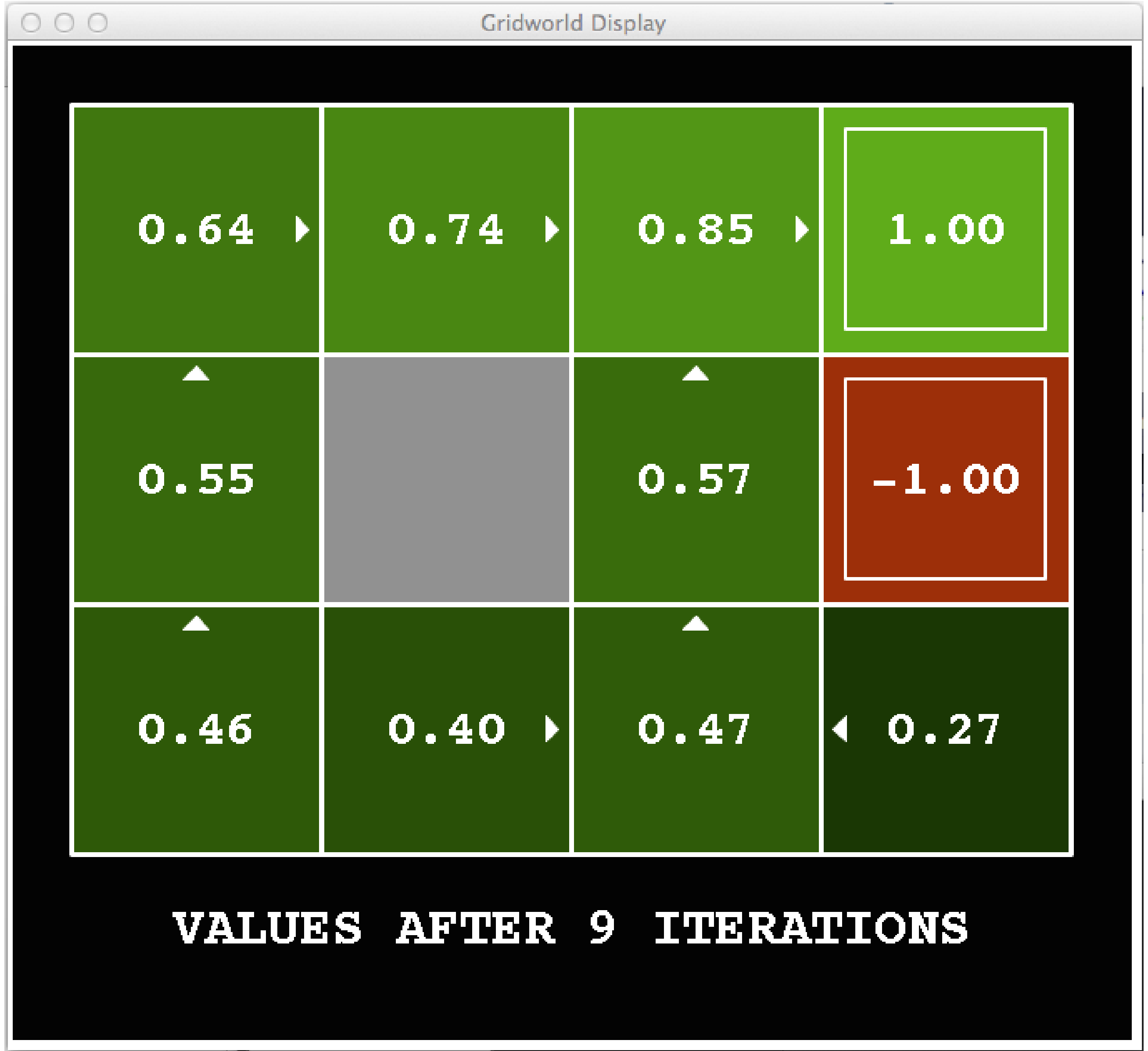
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



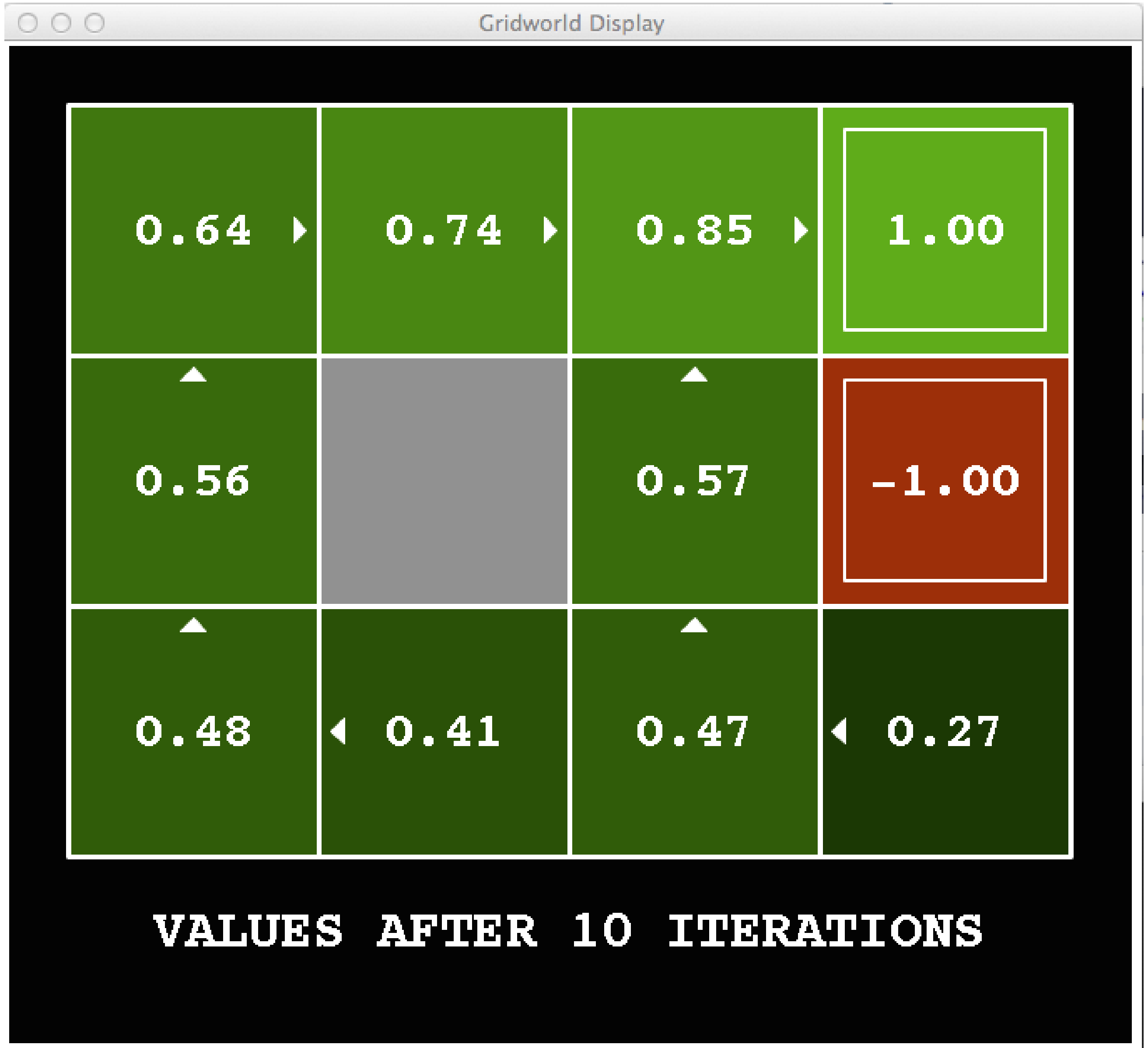
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



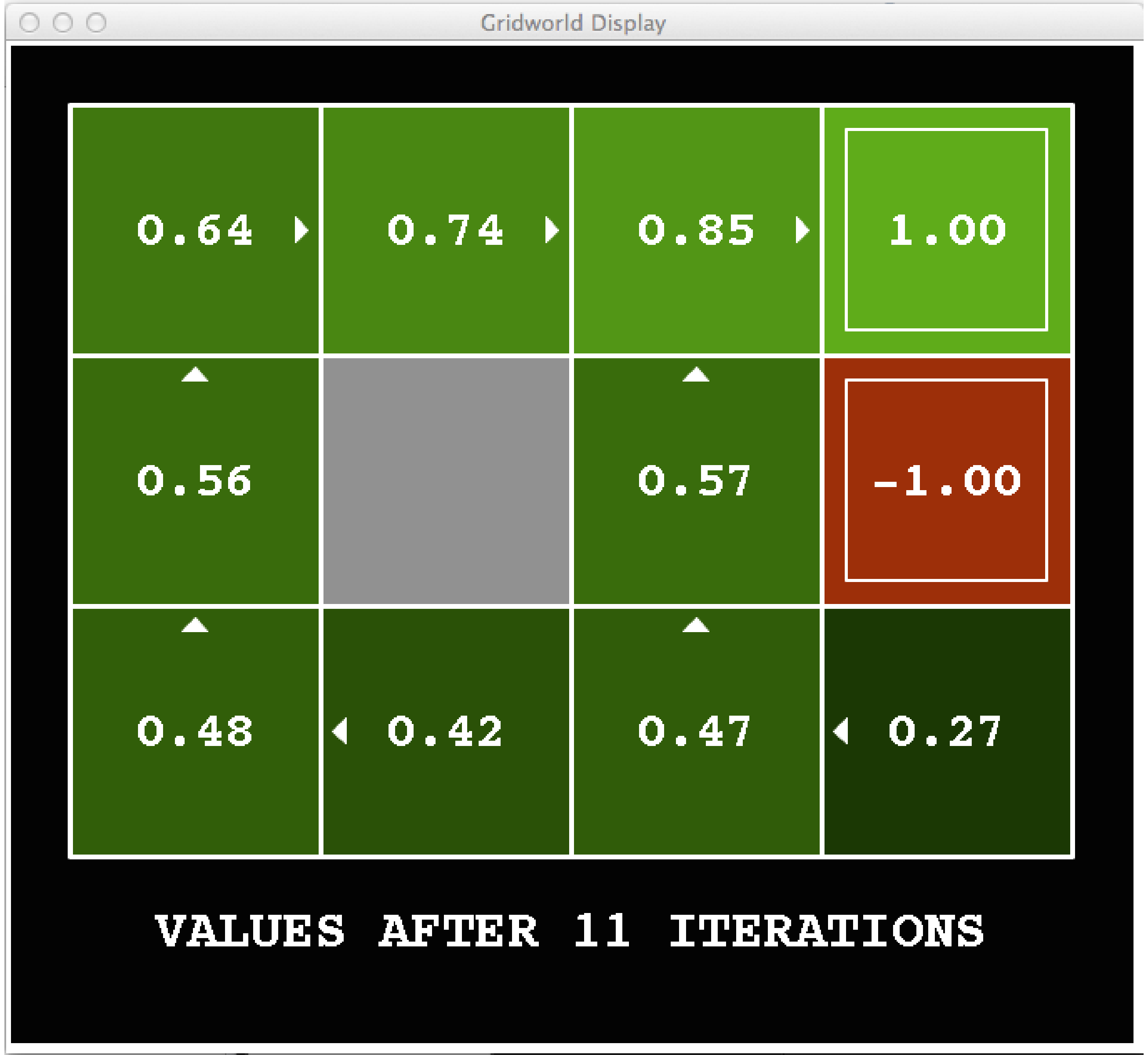
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



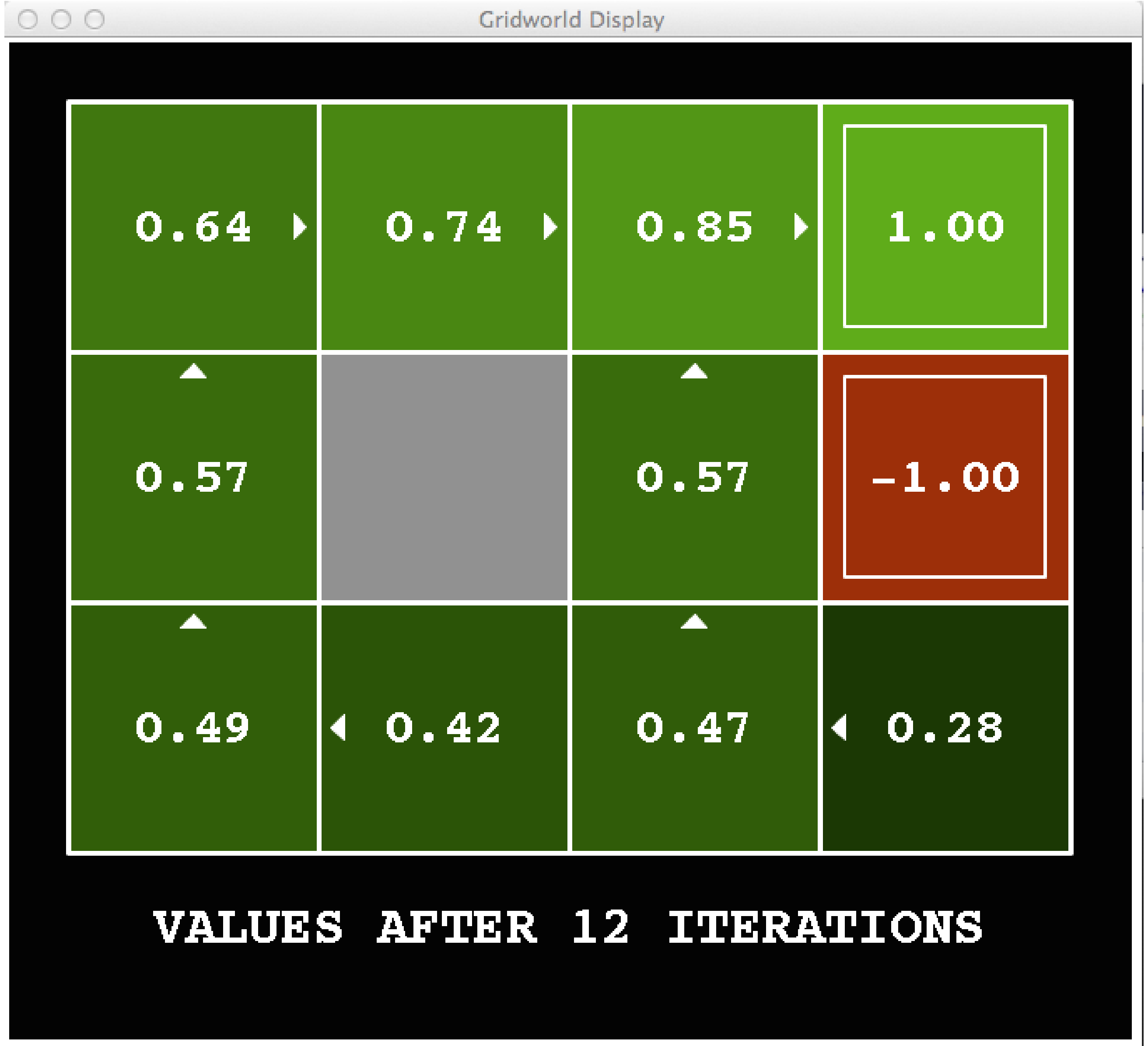
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



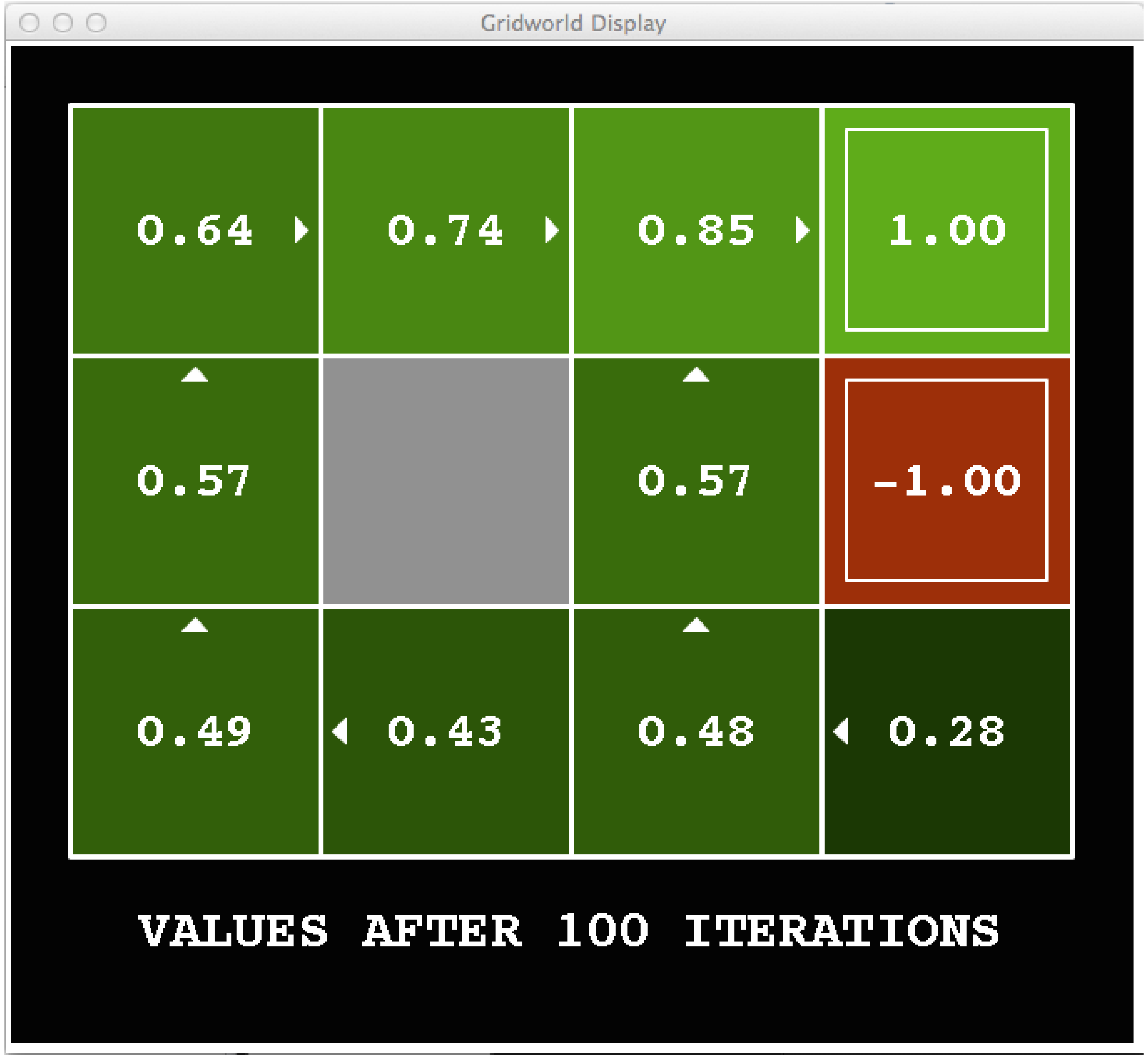
Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



Noise = 0.2
Discount = 0.9
Living reward = 0

Value Iteration



Noise = 0.2
Discount = 0.9
Living reward = 0