University of Ljubljana
Faculty of Electrical Engineering

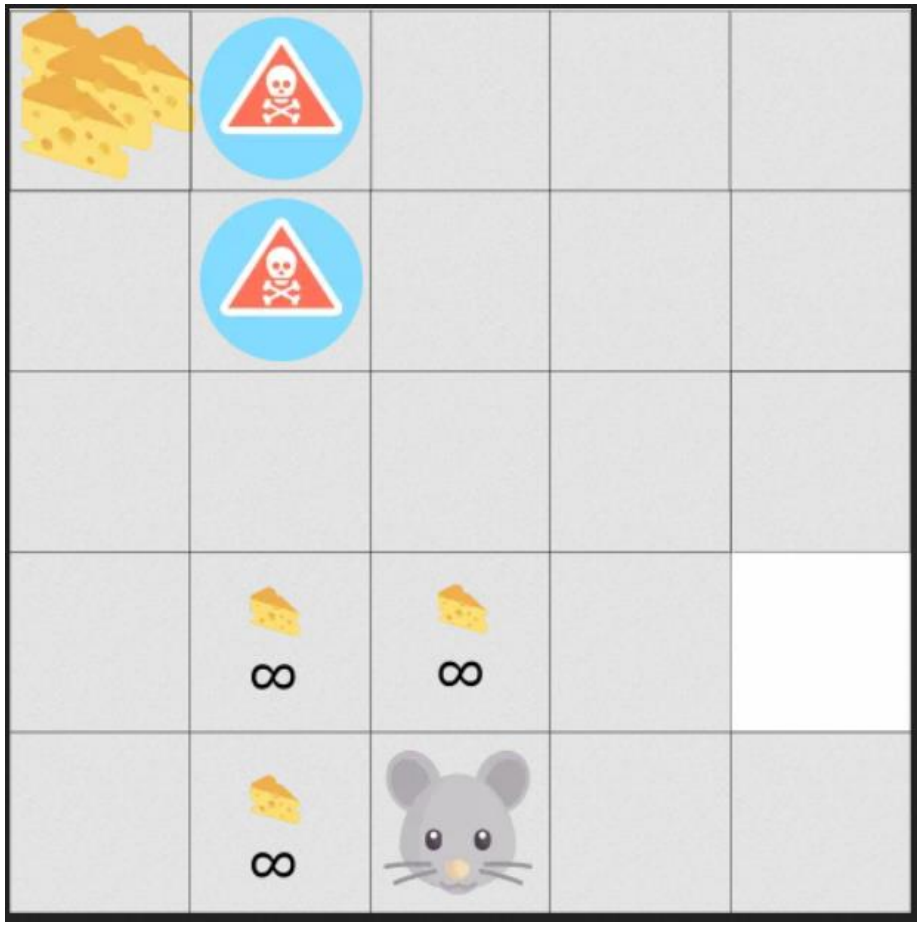
COLLABORATIVE
ROBOTICS CENTRE

robolab

Spodbujevalno učenje – domača naloga

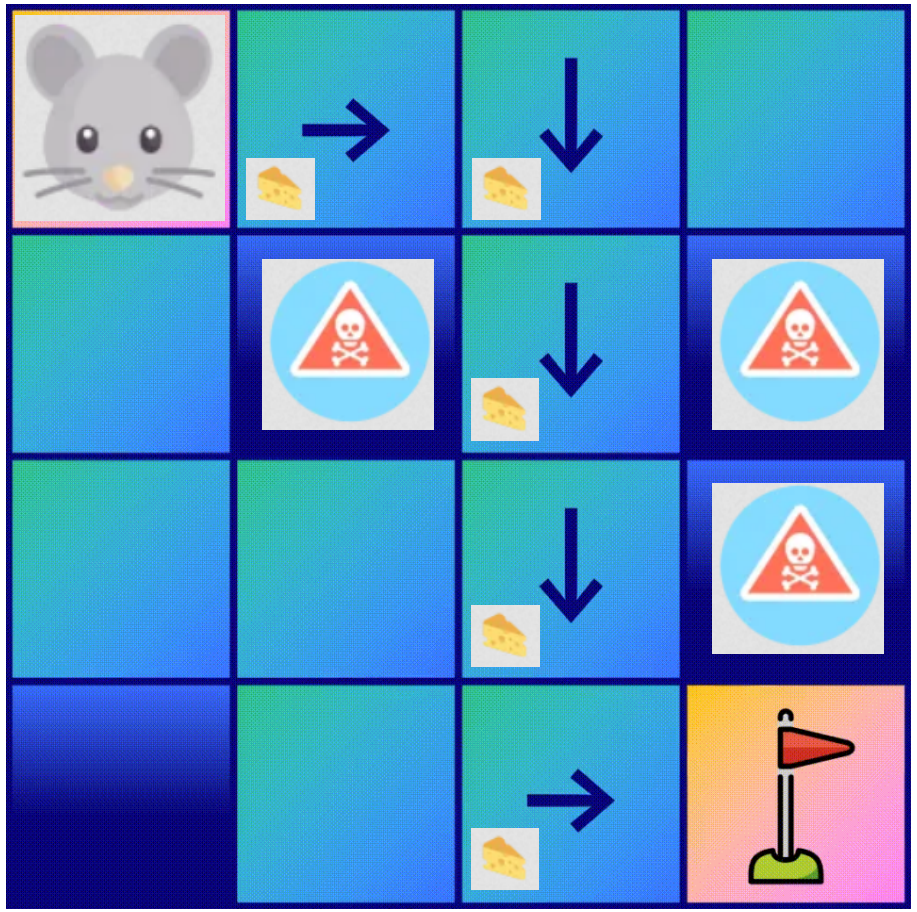
Primer RL: „Miš išče hrano“

- Začetno stanje - S
- Končno stanje - G
- Dovoljeno mesto - F
- Strup – H
- Akcije
  - Levo
  - Dol
  - Desno
  - Gor



Primer RL: „Miš išče hrano“

- Začetno stanje - S
- Končno stanje - G
- Dovoljeno mesto - F
- Strup – H
- Akcije
  - Levo
  - Dol
  - Desno
  - Gor



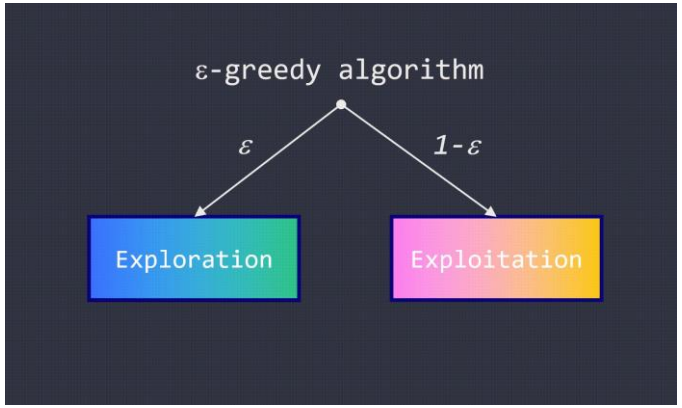
Primer „Miš išče hrano“

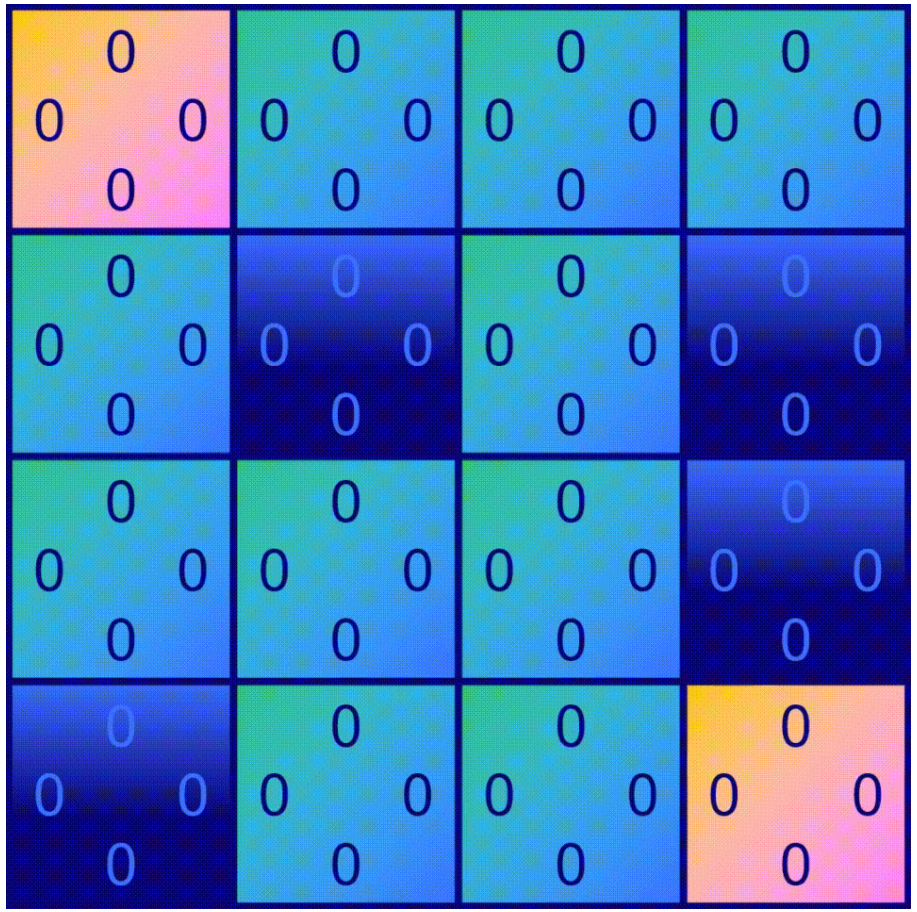
- Reševanje s Q tabelo
  - vrstic: n*n
  - stolpcev: število akcij

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

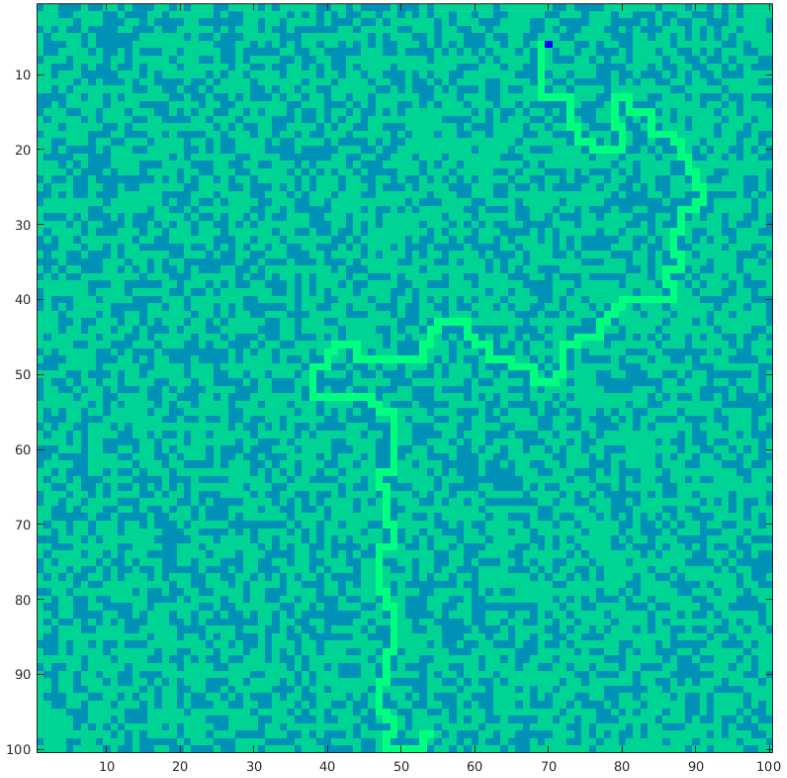
$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha [R_{t+1} + \gamma\, Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

- Izbira akcij

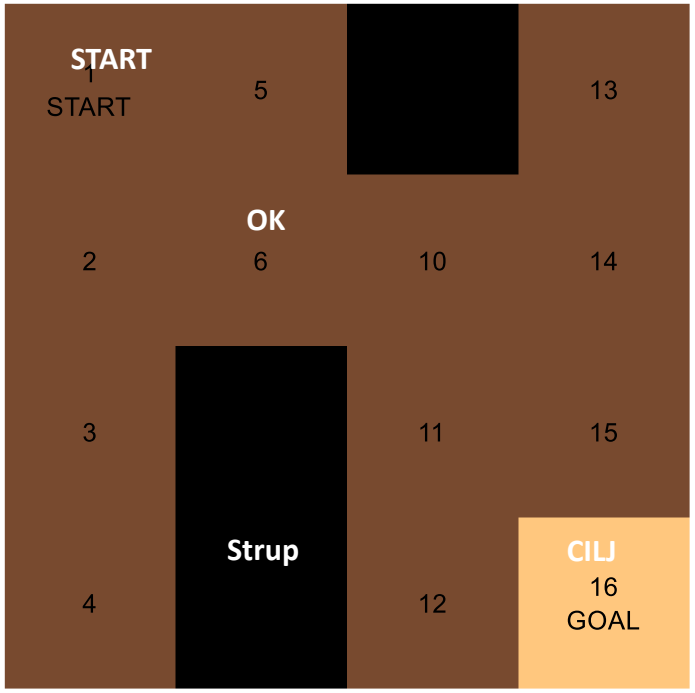


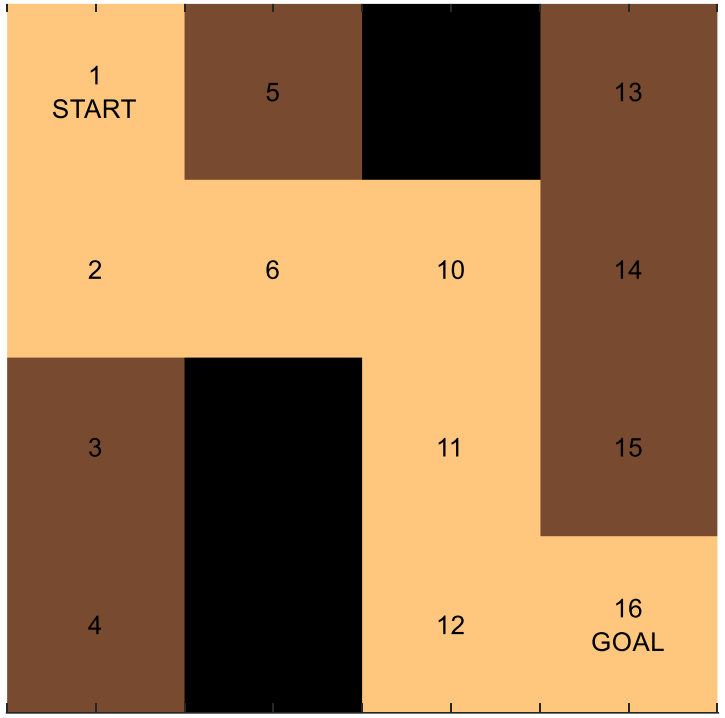


Primer „Miš išče hrano“



Primer „Miš išče hrano“ dimenzije 4 x 4





rešitev problema

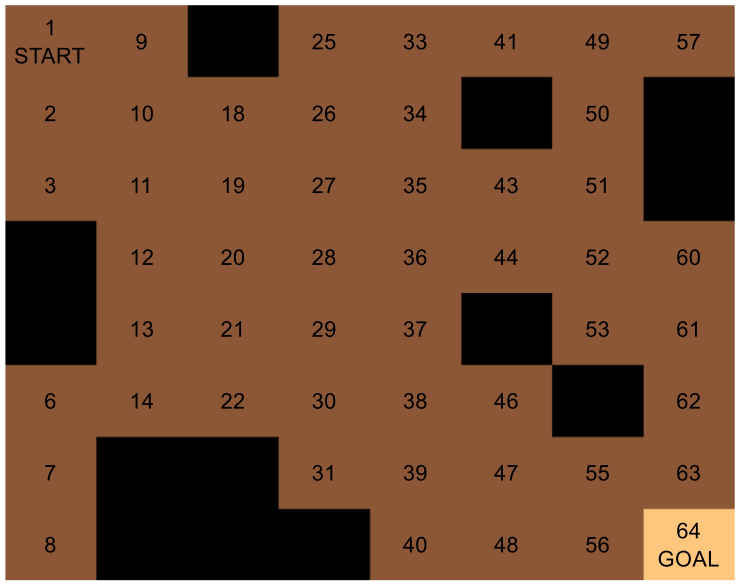
AKCIJE

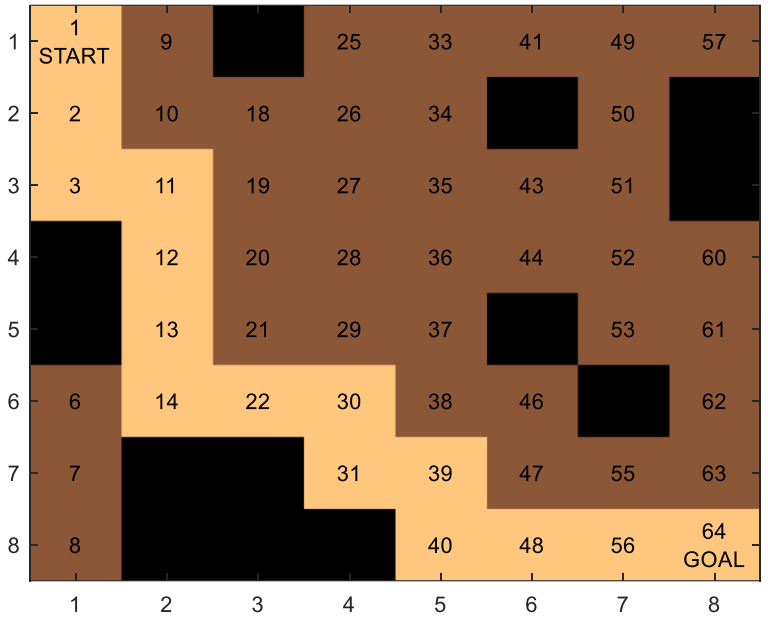
STANJA

	LEFT	DOWN	RIGHT	UP
1.0000	-2.3578	-1.4293	-1.4293	-2.3578
2.0000	-1.4293	-2.3578	-0.4519	-2.3578
3.0000	-2.3578	-4.0000	-4.0000	-1.4293
4.0000	0	0	0	0
5.0000	-2.3578	-0.4519	-0.4519	-1.4293
6.0000	-1.4293	-4.0000	0.5770	-1.4293
7.0000	0	0	0	0
8.0000	0	0	0	0
9.0000	-1.4293	0.5770	-4.0000	-0.4519
10.0000	-0.4519	1.6600	1.6600	-0.4519
11.0000	-4.0000	2.8000	2.8000	0.5770
12.0000	-4.0000	2.8000	4.0000	1.6600
13.0000	0	0	0	0
14.0000	0.5770	2.8000	1.6600	-4.0000
15.0000	1.6600	4.0000	2.8000	1.6600
16.0000	0	0	0	0

Q tabela

Primer „Miš išče hrano“ dimenzije 8 x 8





STANJA

AKCIJE

	LEFT	DOWN	RIGHT	UP
1.0000	-6.3451	-5.6264	-5.6264	-6.3451
2.0000	-5.6264	-8.0000	-4.8699	-6.3451
3.0000	0	0	0	0
4.0000	0	0	0	0
5.0000	-4.9430	-4.9032	-5.0118	-7.9491
6.0000	-4.5211	-4.5135	-7.8540	-4.6217
7.0000	-4.2330	-4.1831	-4.1205	-4.2053
8.0000	-4.0274	-4.0565	-6.7992	-4.0545
9.0000	-6.4069	-4.8699	-4.8923	-5.6488
10.0000	-5.6264	-8.0000	-4.0736	-5.6264
11.0000	0	0	0	0
12.0000	-7.9959	-5.6764	-2.3530	-7.9997
13.0000	-5.2995	-7.8378	-7.9629	-4.7347
14.0000	0	0	0	0

...

Q tabela

Algoritem učenja

Vhod: $strategija\ \pi, uint\ no\_epizod, faktor\ učenja\ \alpha, potek\ \epsilon$
Izhod: optimalna $funkcija\ vrednosti\ Q$ ($če\ je\ število\ epizod\ dovolj\ veliko$)

Inicializacija: $Q(S,A) = 0\ \forall\ S \in \mathcal{S} \wedge A \in \mathcal{A}$, in $Q(končno\ stanje, \cdot) = 0$
for $i = 1$ to no_epizod
    $Izberemo\ vrednost\ \epsilon$
    $Opazujemo\ stanje\ S_0$
    $t \leftarrow 0$
    repeat
        $Izberemo\ akcijo\ A_t\ na\ osnovi\ strategije\ iz\ Q\ (na\ primer\ \epsilon - požrešna\ strategija)$
        $Izvedemo\ akcijo\ A_t\ in\ opazujemo\ nagrado\ R_{t+1}\ in\ novo\ stanje\ S_{t+1}$
        $Posodobimo\ \ Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$
        $t \leftarrow t + 1$
    until $S_t\ je\ končno\ stanje$
end

Parametri za učenje
no_epizod – število epizod učenja
α - parameter hitrosti učenja
γ - parameter zniževanja vrednosti nagrade
ϵ – izbira ϵ-požrešne strategije

Algoritem učenja

Vhod: $strategija\ \pi, uint\ no\_epizod, faktor\ učenja\ \alpha, potek\ \epsilon$
Izhod: optimalna $funkcija\ vrednosti\ Q$ ($če\ je\ število\ epizod\ dovolj\ veliko$)

Inicializacija: $Q(S,A) = 0\ \forall\ S \in \mathcal{S} \wedge A \in \mathcal{A}$, in $Q(končno\ stanje, \cdot) = 0$
for $i = 1$ to no_epizod
    $Izberemo\ vrednost\ \epsilon$
    $Opazujemo\ stanje\ S_0$
    $t \leftarrow 0$
    repeat
        $Izberemo\ akcijo\ A_t\ na\ osnovi\ strategije\ iz\ Q\ (na\ primer\ \epsilon - požrešna\ strategija)$
        $Izvedemo\ akcijo\ A_t\ in\ opazujemo\ nagrado\ R_{t+1}\ in\ novo\ stanje\ S_{t+1}$
        $Posodobimo\ \ Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma\ Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$
        $t \leftarrow t + 1$
    until $S_t\ je\ končno\ stanje$
end

Parametri za učenje
no_epizod – število epizod učenja
α - parameter hitrosti učenja
γ - parameter zniževanja vrednosti nagrade
ϵ – izbira ϵ-požrešne strategije

Matlab predloga

- predloga *frozen_lake_tmplt.m*
  - v spremenljivko *vpisna_stevilka* vpisete vašo vpisno številko in dodate še eno cifro
  - ustvari tabelo *lake* s frozen lake okoljem in nagradami
  - prikaz okolja
  - **vpišete vaš algoritem za spodbujevalno učenje**
  - na koncu skripte še klic funkcije *visualization_Q4.p* za prikaz končne rešitve
  - zapis *num_steps = visualization_Q_arrows4(Q, lake)* izriše akcije v obliki puščic

```
%% Create env
vpisna_stevilka = 649901670;
rng(vpisna_stevilka)
n = 4;

klet = -1*ones(n,n);

for i=1:n
    for j=1:n
        if (rand() < 0.25)
            klet(i,j) = -n;
        end
    end
end
klet(1,1) = -1;
klet(1,2) = -1;
klet(2,1) = -1;
klet(2,2) = -1;
klet(n-1,n-1) = -1;
klet(n,n-1) = -1;
klet(n-1,n) = -1;
klet(n,n) = n;

% Render environment
disp(klet)

fh = figure;
imagesc(klet);
colormap(copper);

for i=1:n
    for j=1:n

        if (i==1) && (j == 1)
            text(1,1,{'1','START'},'HorizontalAlignment','center');
        elseif (i==n) && (j==n)
            text(n,n,{num2str(n*n),'GOAL'},'HorizontalAlignment','center')
        else
            text(j,i,num2str(i+n*(j-1)),'HorizontalAlignment','center')
        end
    end
end

axis off

%%
%Vaša koda


%%
% Vizualizacija rešitve
indexQ = int32([(1:(n*n))]');
visQ = table(indexQ,Q)

num_steps = vizualizacija_Q4(Q, klet);
num_steps = visualization_Q_arrows4(Q, klet);
```

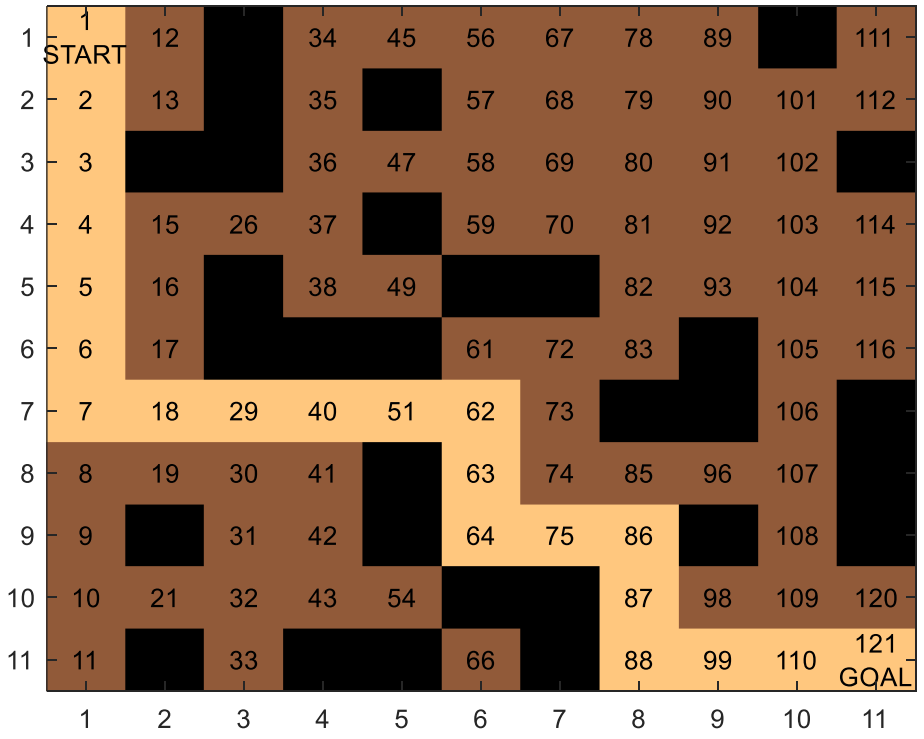
Matlab predloga

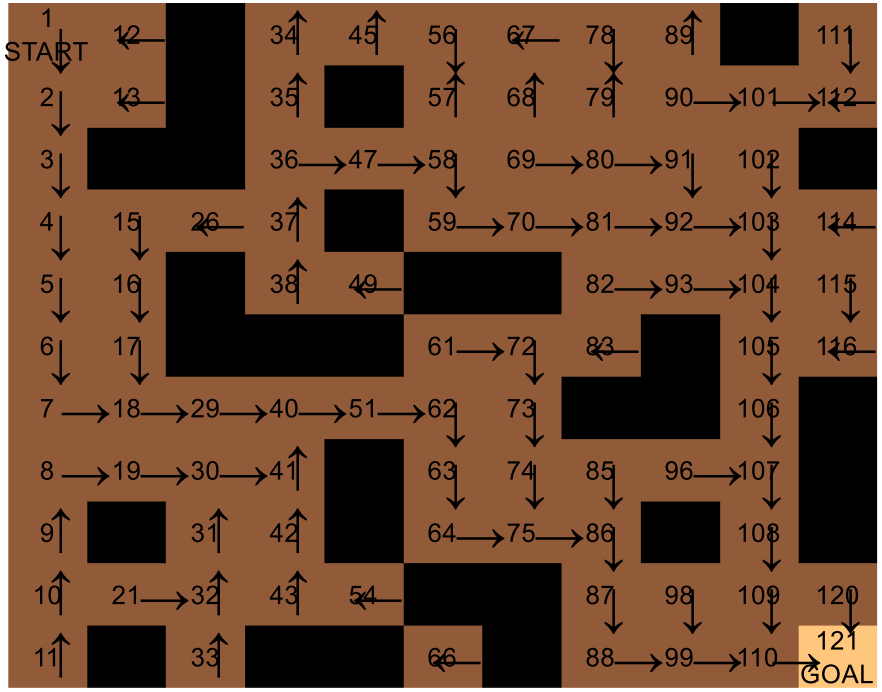
- vizualizacija rešitve s funkcijo *visualization_Q4.p*
- Q tabela mora imeti n x n vrstic, ter 4 stolpce za 4 akcije:
  - 1. stolpec za akcijo LEFT
  - 2. stolpec za akcijo DOWN
  - 3. stolpec za akcijo RIGHT
  - 4. stolpec za akcijo UP

- vizualizacija rešitve s funkcijo *visualization_Q5.p*
- Q tabela mora imeti n x n vrstic, ter 5 stolpcev za 5 akcij:
  - 1. stolpec za akcijo LEFT
  - 2. stolpec za akcijo DOWN
  - 3. stolpec za akcijo RIGHT
  - 4. stolpec za akcijo UP
  - 5. stolpec za akcijo RIGHT-DOWN

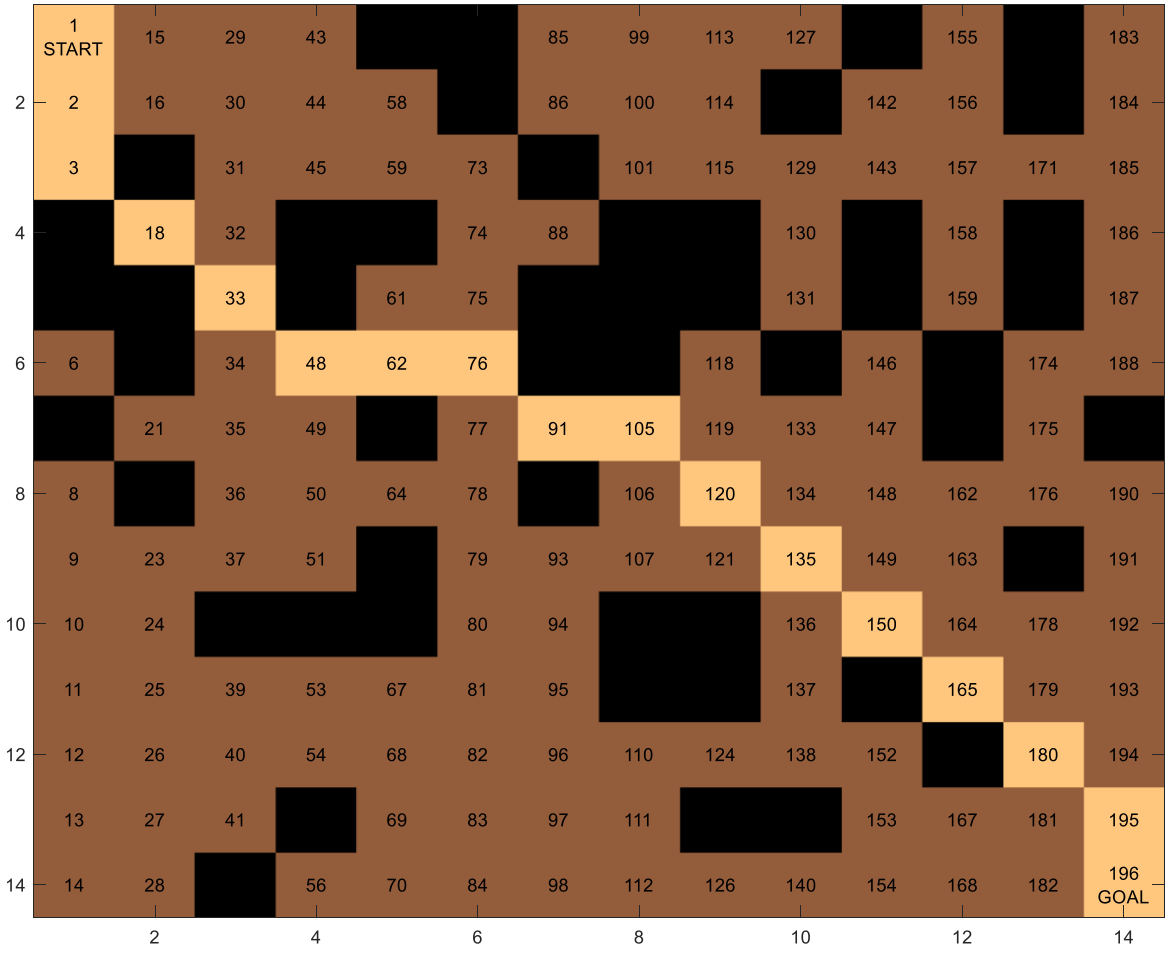
```
49
50  %%
51  % Vizualizacija rešitve
52  indexQ = int32([(1:(n*n))]');
53  visQ = table(indexQ,Q)
54
55  num_steps = vizualizacija_Q4(Q, klet);
56  num_steps = visualization_Q_arrows4(Q, klet);
57
```

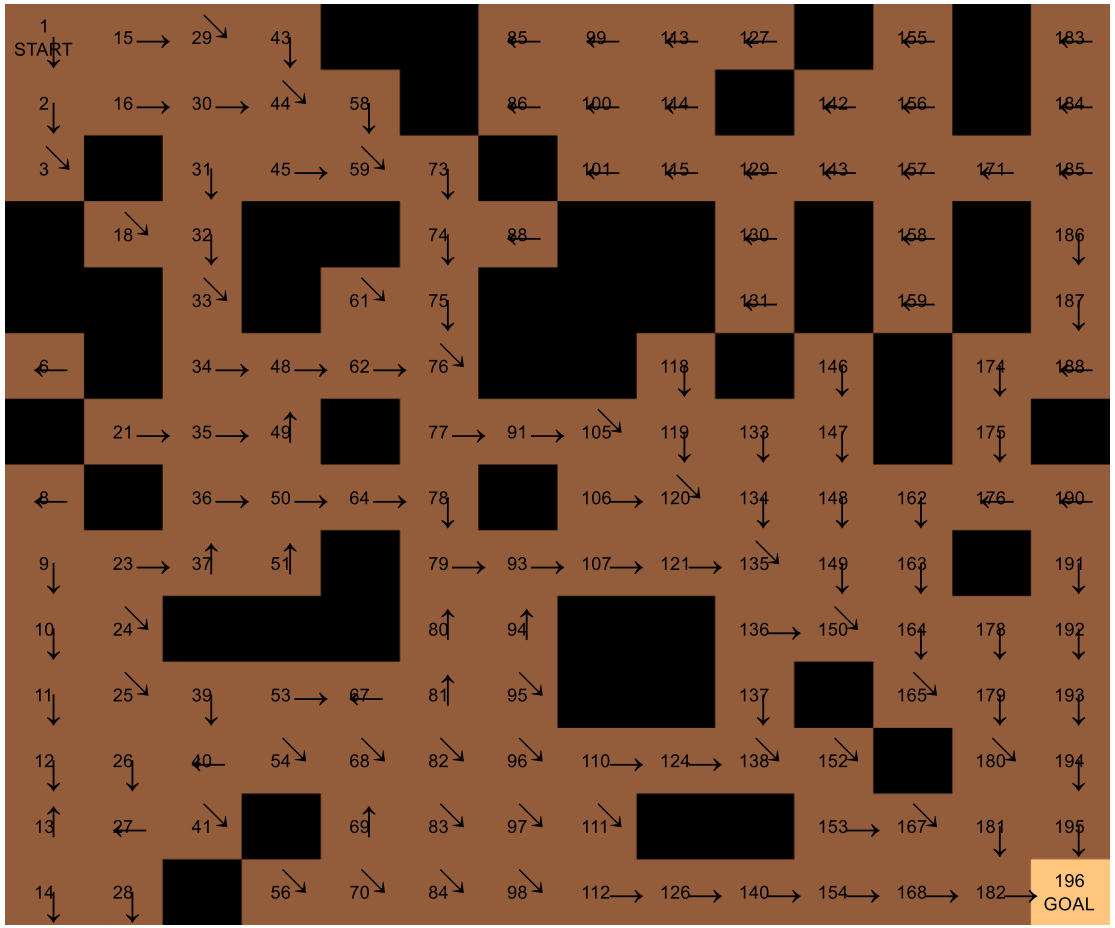
Primeri rešitev





Primeri rešitev





Vrednosti stanj

- Predavanje
  - Ovrednotenje naključne strategije v „majhni mreži“ (stran 4, zgornja prosojnica)
  - Deterministično iteriranje vrednosti (stran 5 , spodnja prosojnica)
  - *11 - Spodbujevalno učenje - planiranje in predikcija.pdf*