

بسم الله الرحمن الرحيم



محیط ناشناخته

استاد مربوطه: استاد حسین کارشناس

دستیار آموزشی: پوریا صامتی

اعضای تیم

یونس ایوبی راد ۴۰۱۳۶۱۳۰۱۱

پویا اسفندانی ۴۰۱۳۶۱۳۰۰۵

معرفی تابع ها

```
def get_index(pig):  
    number = int(''.join('1' if x else  
    '0' for x in pig), 2)  
    return number  
  
def get_state(state):  
    return state[1]*8+state[0]
```

در این قسمت دو تابع معرفی شده که تابع اول خوک ها را میگیرد و آنها را به یک عدد ۸ بیتی تبدیل می کند مثلا اگر تمام خوک ها زنده باشند مقدار ۲۵۵ را بر می گرداند.

تابع `get_state` این تابع مختصات x و y را به یک عدد یکتا تبدیل می کند. با ضرب مقدار y در ۸ و اضافه کردن x ، یک مقدار عددی منحصر به فرد به دست می آید.

معرفی مقادیر اولیه

```
pig = 256
num_states = 64
num_actions = 4
q_table = np.zeros((pig, num_states,
num_actions), dtype=np.int64)
learning_rate = 0.05
discount_factor = 0.99
epsilon = 1.0
epsilon_decay = 0.9997
epsilon_min = 0.1
episode_reward = []
```

این کد پارامترهای الگوریتم یادگیری تقویتی را تعریف می‌کند. تعداد موجودیت‌ها یا محیط‌ها را مشخص می‌کند. num_states. تعداد حالت‌های ممکن در محیط است و num_actions تعداد اقدامات ممکن در هر حالت را نشان می‌دهد. q_table جداولی برای ذخیره مقادیر Q هستند. learning_rate نرخ یادگیری است که تاثیر تجربه‌های جدید بر مدل را تنظیم می‌کند و discount_factor مقدار توجه مدل به جوایز بلندمدت را مشخص می‌کند. epsilon مقدار جست‌وجوی تصادفی (exploration) مدل است که با epsilon_decay به تدریج کاهش می‌یابد و به حداقل epsilon_min می‌رسد. در نهایت، episode_reward برای ذخیره جوایز هر اپیزود استفاده می‌شود.

معرفی کلاس main

```
previous_pig = get_index(pigs)
previous_state_index = get_state(state)
if np.random.rand() < epsilon:
    action = np.random.choice(num_actions)
else:
    action =
np.argmax(q_table[previous_pig,previous_state_index, :])

next_state, reward, pig_state, done =
env.step(action)
next_pig = get_index(pig_state)
next_state_index = get_state(next_state)
```

در قسمت اول از کد خوک های قبلی و استیت سابق به ایندکس تبدیل می شوند و برای استفاده در q table آماده می شوند.

در قسمت بعدی از ϵ -greedy استفاده می شود، روش کار این کد به این گونه است که یک عدد رندوم تولید می کند اگر این عدد کمتر از اپسیلون بود به صورت رندوم در محیط حرکت میکند و در غیر این صورت ایندکس ماکسیموم مقدار Q-table را برمی گرداند، سپس بعد از به دست آوردن اکشن آن را در محیط اعمال میکند، سپس بر اساس تابع های معرفی شده استیت های جدید را محاسبه می کند.

قسمت یادگیری کد

```
q_table[previous_pig, previous_state_index,  
action] = (q_table[previous_pig,  
previous_state_index, action] + learning_rate *  
(reward + discount_factor *  
np.max(q_table[next_pig, next_state_index, :]) -  
q_table[previous_pig, previous_state_index,  
action]))  
pigs = pig_state  
state = next_state  
total_reward += reward
```

در قسمت اول این کد با استفاده از فرمول q learning مقادیر q-table را بروزرسانی می‌کند، سپس مقادیر خوک‌ها و استیت جدید را جایگزین مقادیر قبلی می‌کند و به حلقه ادامه می‌دهد.

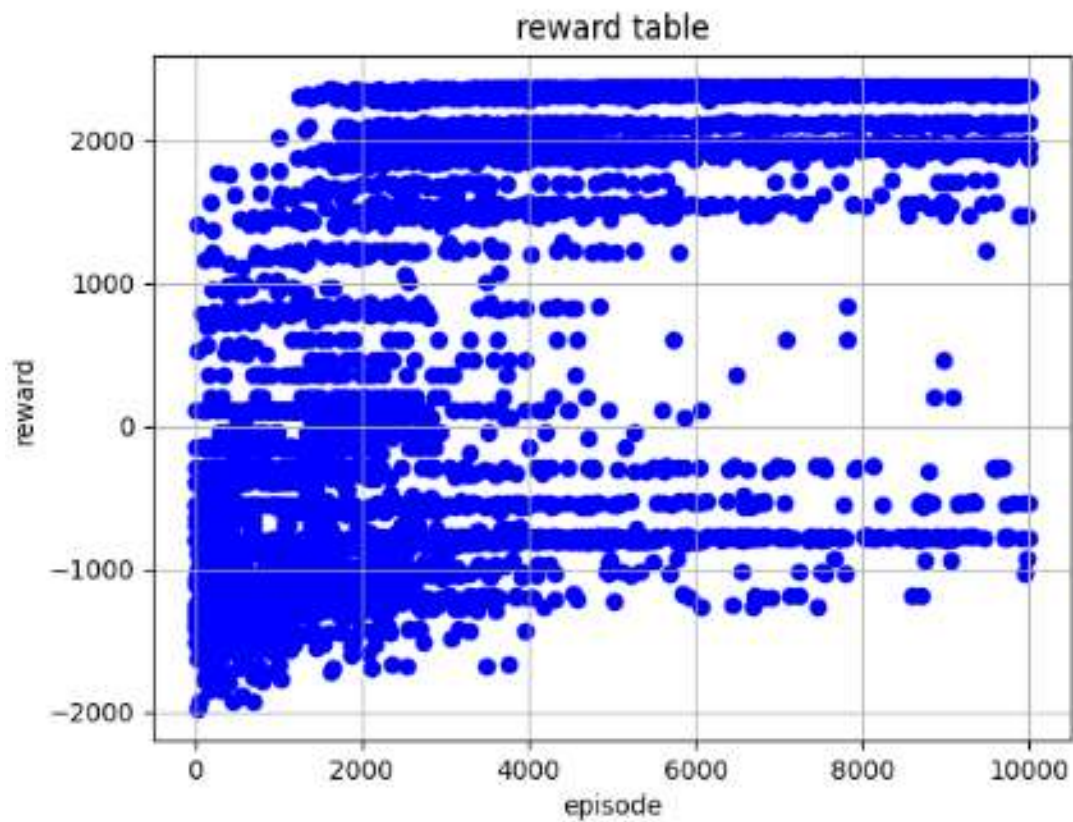
قسمت پایان اپیزود

```
if done:
    print(f"Episode {episode + 1} finished
with reward: {total_reward}")
    if episode == 9995:
        epsilon = 0
        epsilon_min = 0
    episode_reward.append(total_reward)
    running = False

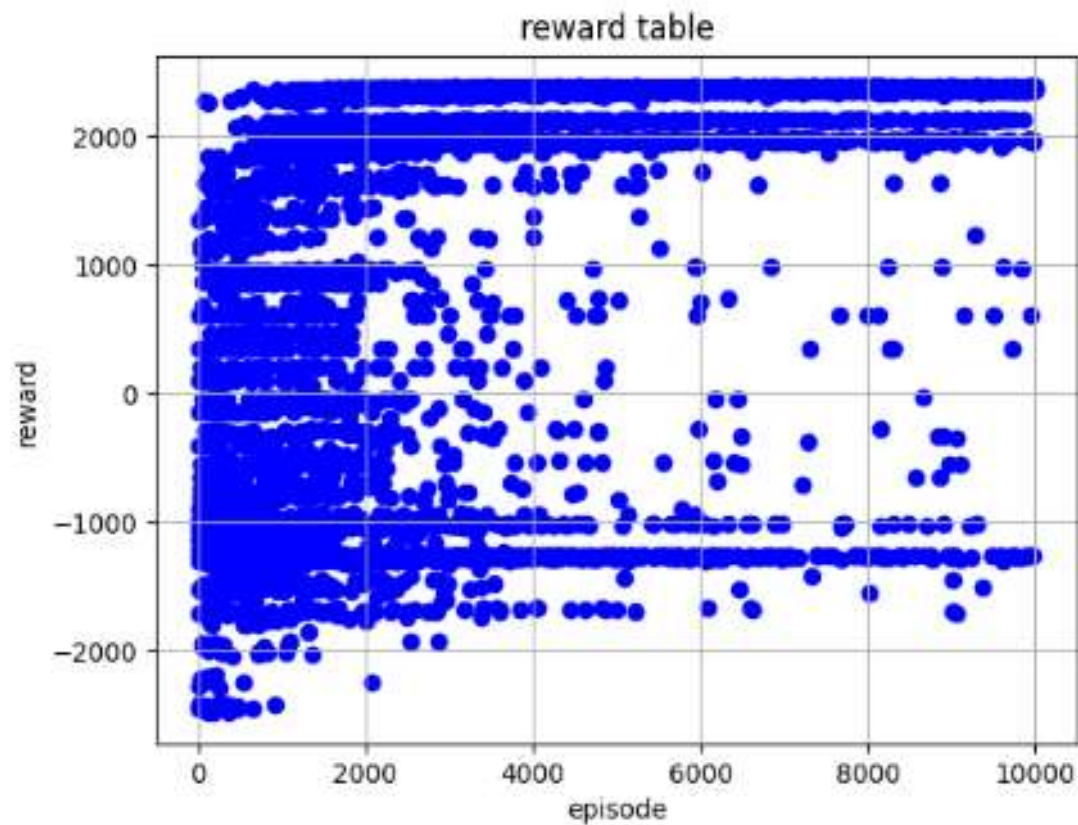
pygame.display.flip()
epsilon = max(epsilon_min, epsilon *
epsilon_decay)
```

در قسمت اول اگر کار مرحله تمام شده بود ریوارد را صفر میکند و بازی را خاتمه می‌دهد و اگر اپیزود ها برابر با ۹۹۵۰ بودند دیگر به صورت رندوم تصمیم نمی‌گیرد و سپس مقدار اپسیلون را کاهش می‌دهد.

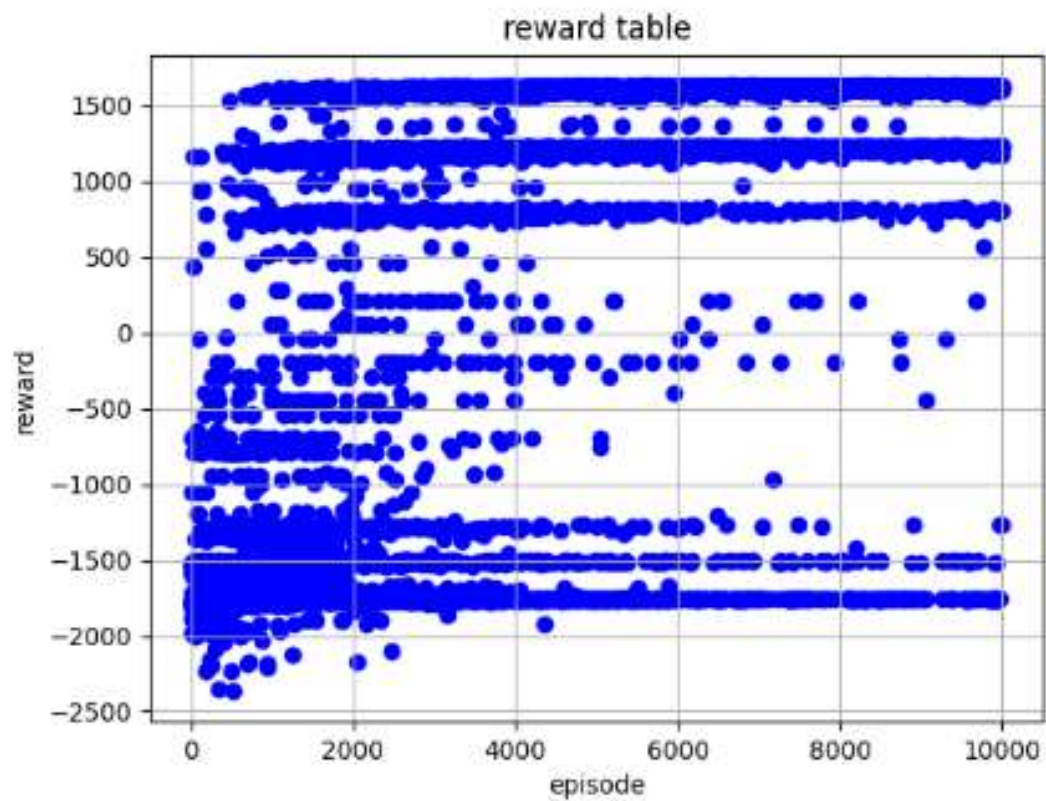
نمونه کار کد



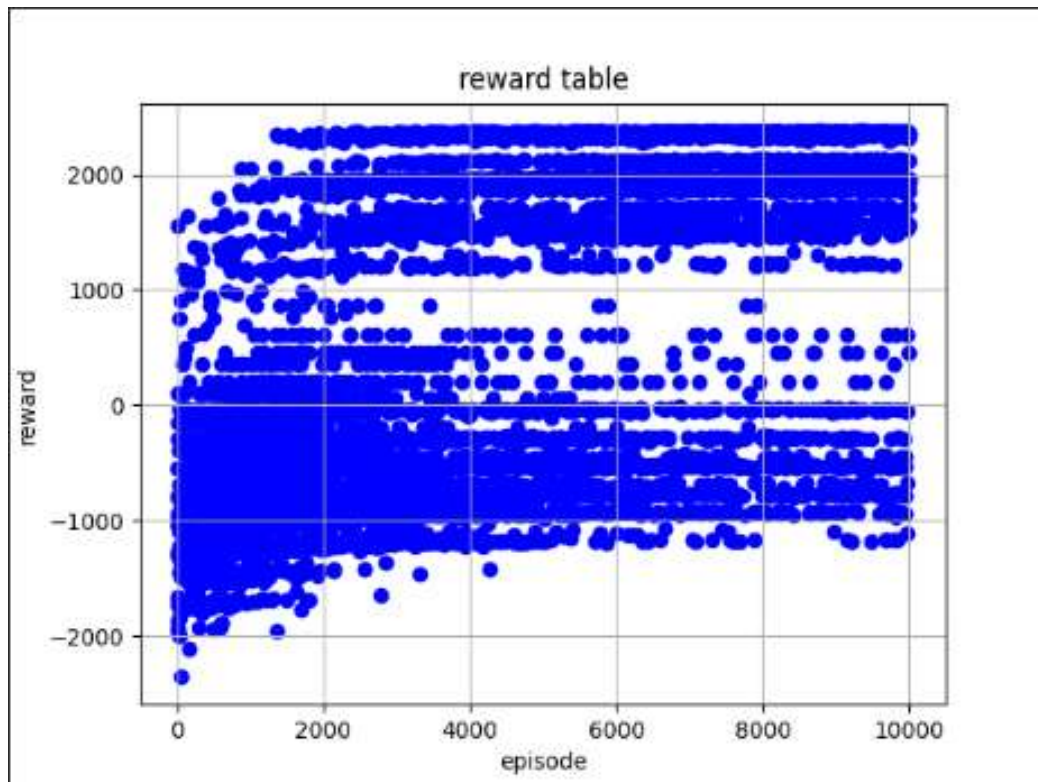
```
Episode 9996 finished with reward: 2352
Episode 9997 finished with reward: 2363
Episode 9998 finished with reward: 2367
Episode 9999 finished with reward: 2364
Episode 10000 finished with reward: 2348
```



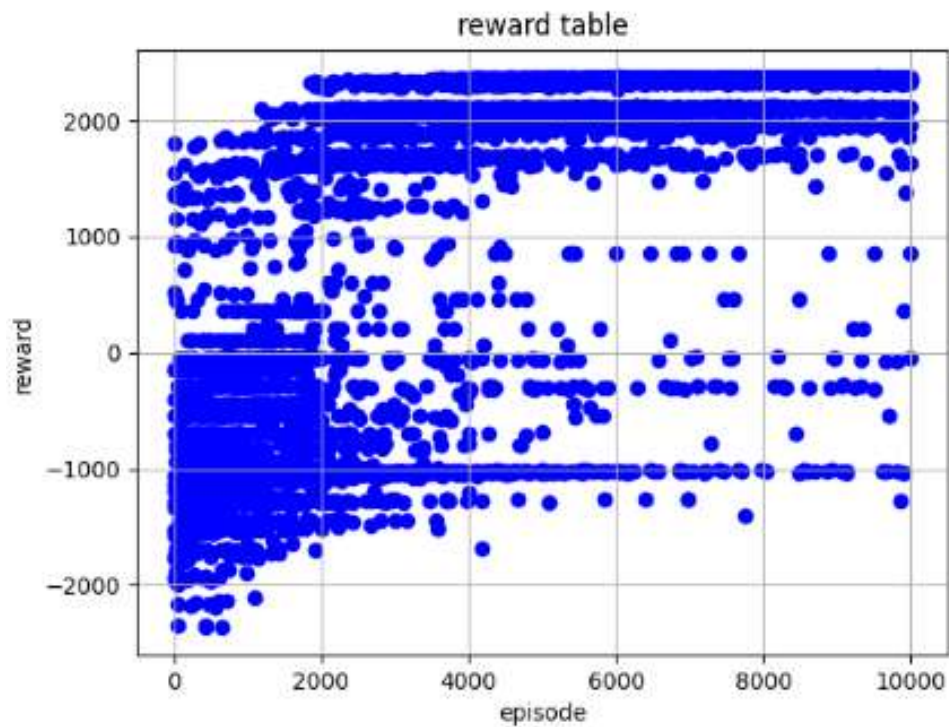
```
Episode 9995 finished with reward: 2375  
Episode 9996 finished with reward: 2378  
Episode 9997 finished with reward: 2383  
Episode 9998 finished with reward: 2388  
Episode 9999 finished with reward: 2381  
Episode 10000 finished with reward: 2385
```

```
Episode 9995 finished with reward: 1623
Episode 9996 finished with reward: 1182
Episode 9997 finished with reward: 1622
Episode 9998 finished with reward: 1609
Episode 9999 finished with reward: 1635
Episode 10000 finished with reward: 1628
```



```
Episode 9995 finished with reward: 1937
Episode 9996 finished with reward: 1959
Episode 9997 finished with reward: 1849
Episode 9998 finished with reward: 2368
Episode 9999 finished with reward: 1927
Episode 10000 finished with reward: 2126
```



```
Episode 9995 finished with reward: 2351
Episode 9996 finished with reward: 2348
Episode 9997 finished with reward: 2361
Episode 9998 finished with reward: 2356
Episode 9999 finished with reward: 2362
Episode 10000 finished with reward: 2361
```