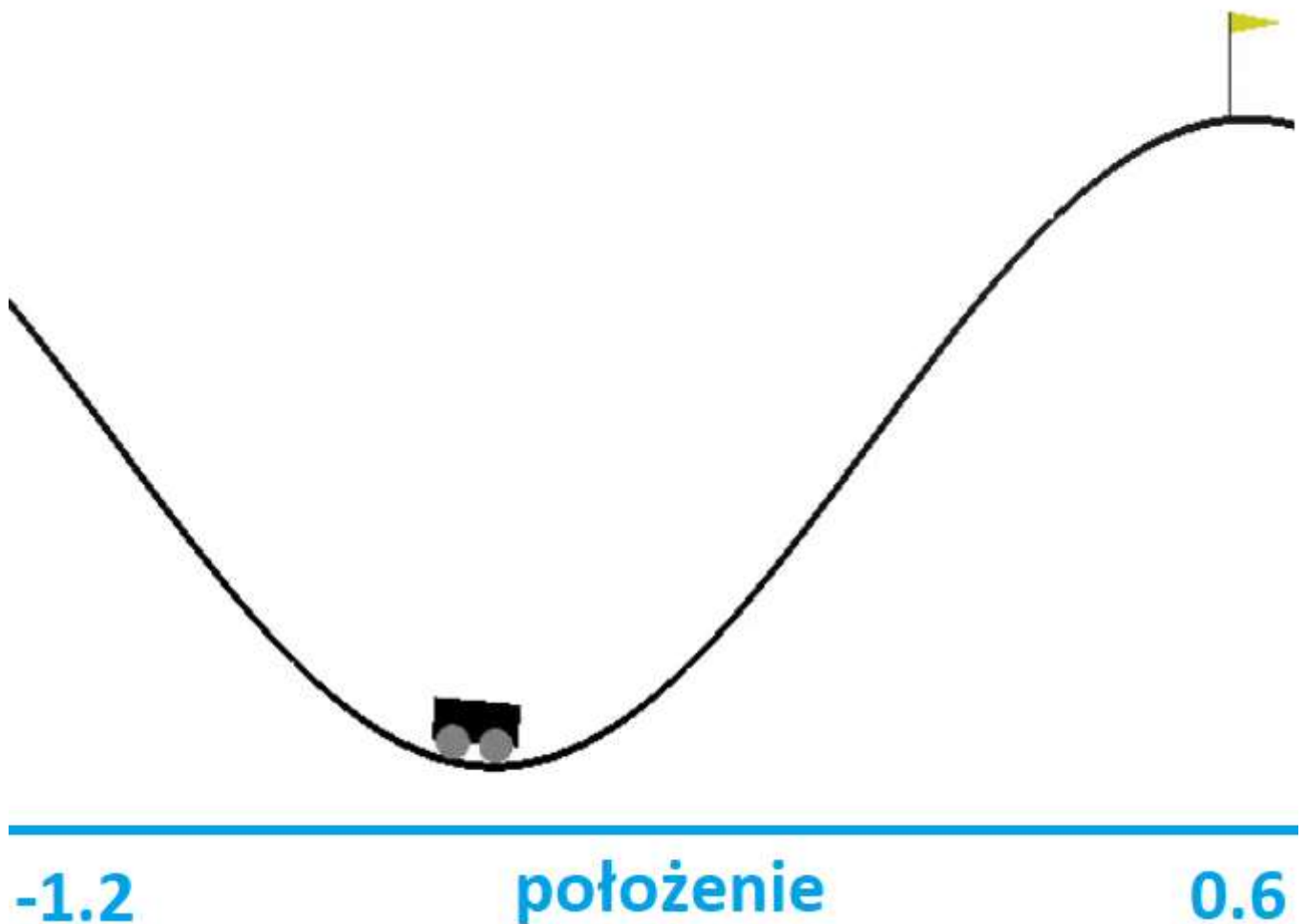


## ▼ Środowisko MountainCar

Pojazd znajduje się na **jednowymiarowym torze** (patrz rysunek poniżej), umieszczonym pomiędzy dwoma wzniesieniami. **Celem jest wjechać pojazdu na górę po prawej stronie** (miejsce oznaczone chorągiewką). Silnik samochodu nie jest wystarczająco mocny, aby bezpośrednio wjechać na górę. **Możliwym rozwiązaniem jest jazda "tam i z powrotem", w lewo i prawo i nabranie rozpędu.**



```
import numpy as np
import gym
import matplotlib.pyplot as plt
```

**Stan** reprezentowany jest za pomocą wektora **[położenie, prędkość]**. Stan początkowy ma losowe **położenie od -0,6 do -0,4** (położenie w dolinie) i **prędkość=0**:

```
env = gym.make("MountainCar-v0")
State_0 = env.reset()
State_0
```

```
array([-0.50484306,  0.          ])
```

Wartości **min** i **max** położenia oraz prędkości:

```
print("położenie min=",env.observation_space.low[0]," max=",env.observation_space.high[0])
print("prędkość min=",env.observation_space.low[1]," max=",env.observation_space.high[1])

położenie min= -1.2   max= 0.6
prędkość min= -0.07  max= 0.07
```

## ▼ Akcje

Możliwe do wykonania są **3 akcje**, oznaczone **0,1,2**:

```
env.action_space

Discrete(3)
```

Przetestujmy **akcję 0** - wykonajmy ją **10 razy**:

```
env.reset()
for _ in range(10):
    action = 0
    obs, reward, done, info = env.step(action)
    print("stan=",obs)

stan= [-5.98463955e-01 -4.46484837e-04]
stan= [-0.59935366 -0.0008897 ]
stan= [-0.60068008 -0.00132642]
stan= [-0.60243353 -0.00175345]
stan= [-0.60460122 -0.00216769]
stan= [-0.60716735 -0.00256613]
stan= [-0.61011326 -0.00294591]
stan= [-0.61341757 -0.00330431]
stan= [-0.61705636 -0.00363879]
stan= [-0.62100337 -0.00394701]
```

Jak widać powyżej akcja 0 powoduje **pchnięcie pojazdu w lewą stronę** (zmniejszeniu ulega pierwsza współrzędna) i **zwiększeniu ulega prędkość pojazdu w lewą stronę** (druga współrzędna - ujemna wartość prędkości).

W przypadku pozostałych akcji: **1 - brak pchnięcia pojazdu, 2 - popchnięcie w prawo**.

**Przetestujmy działanie środowiska - wykonajmy pewną ilość (dokładnie 50) akcji 0 (pchnięcie**

```
env.reset()
for _ in range(50):
    action = 0
    obs, reward, done, info = env.step(action)
    print("położenie: ",obs[0]," prędkość: ",obs[1])
    if obs[1]<0:
        print("pojazd porusza się w LEWO")
    else:
        print("pojazd porusza się w PRAWO")

położenie: -0.5337527216458521 prędkość: -0.0009308354969562322
pojazd porusza się w LEWO
położenie: -0.5356074143245128 prędkość: -0.0018546926786606574
pojazd porusza się w LEWO
położenie: -0.5383720616934771 prędkość: -0.00276464736896428
pojazd porusza się w LEWO
położenie: -0.5420259456861962 prędkość: -0.003653883992719178
pojazd porusza się w LEWO
położenie: -0.5465416962853357 prędkość: -0.004515750599139494
pojazd porusza się w LEWO
położenie: -0.5518855108093416 prędkość: -0.005343814524005863
pojazd porusza się w LEWO
położenie: -0.5580174293525734 prędkość: -0.006131918543231838
pojazd porusza się w LEWO
położenie: -0.5648916664539041 prędkość: -0.006874237101330716
pojazd porusza się w LEWO
położenie: -0.572456998363301 prędkość: -0.007565331909396902
pojazd porusza się w LEWO
położenie: -0.5806572042889377 prędkość: -0.008200205925636772
pojazd porusza się w LEWO
położenie: -0.5894315587745644 prędkość: -0.008774354485626731
pojazd porusza się w LEWO
położenie: -0.5987153709410042 prędkość: -0.009283812166439837
pojazd porusza się w LEWO
położenie: -0.6084405648128561 prędkość: -0.009725193871851901
pojazd porusza się w LEWO
położenie: -0.6185362934448854 prędkość: -0.010095728632029262
pojazd porusza się w LEWO
położenie: -0.6289295781739718 prędkość: -0.01039328472908644
pojazd porusza się w LEWO
położenie: -0.6395459631626056 prędkość: -0.010616384988633857
pojazd porusza się w LEWO
położenie: -0.6503101745683233 prędkość: -0.010764211405717703
pojazd porusza się w LEWO
położenie: -0.6611467732490478 prędkość: -0.010836598680724506
pojazd porusza się w LEWO
położenie: -0.6719807899475898 prędkość: -0.010834016698541971
pojazd porusza się w LEWO
położenie: -0.6827383324068255 prędkość: -0.010757542459235643
pojazd porusza się w LEWO
położenie: -0.6933471548326096 prędkość: -0.01060882242578406
pojazd porusza się w LEWO
położenie: -0.703737181493554 prędkość: -0.010390026660944518
pojazd porusza się w LEWO
położenie: -0.713840977946657 prędkość: -0.010103796453102961
pojazd porusza się w LEWO
```

```

położenie: -0.7235941653062388 prędkość: -0.009753187359581758
pojazd porusza się w LEWO
położenie: -0.7329357750202546 prędkość: -0.009341609714015927
pojazd porusza się w LEWO
położenie: -0.7418085436704679 prędkość: -0.008872768650213225
pojazd porusza się w LEWO
położenie: -0.7501591492657832 prędkość: -0.00835060559531529
pojazd porusza się w LEWO
położenie: -0.757938392259987 prędkość: -0.007779242994203893
pojazd porusza się w LEWO
położenie: -0.765101326024106 prędkość: -0.007162933764118942

```

Jak widać powyżej pojazd **początkowo porusza się w lewo, ale po pewnym czasie zaczyna opadać, czyli poruszać się w prawo**. W tym momencie prędkość pojazdu zmienia wartość z ujemnej (prędkość w lewo) na dodatnią (prędkość w prawo).

Zobaczmy co się dzieje dalej. Spróbujmy wykonać teraz pewną ilość razy **akcję 1 czyli brak pchnięcia** (UWAGA: **nie resetujemy stanu środowiska!!!**):

```

for _ in range(50):
    action = 1
    obs, reward, done, info = env.step(action)
    print("położenie: ",obs[0]," prędkość: ",obs[1])
    if obs[1]<0:
        print("pojazd porusza się w LEWO")
    else:
        print("pojazd porusza się w PRAWO")

położenie: -0.7297460418484281 prędkość: 0.009989698410582948
pojazd porusza się w PRAWO
położenie: -0.7183069277647243 prędkość: 0.01143911408370376
pojazd porusza się w PRAWO
położenie: -0.7054891406320862 prędkość: 0.012817787132638047
pojazd porusza się w PRAWO
położenie: -0.691373873885804 prędkość: 0.014115266746282312
pojazd porusza się w PRAWO
położenie: -0.6760527543811004 prędkość: 0.015321119504703593
pojazd porusza się w PRAWO
położenie: -0.6596276780729396 prędkość: 0.01642507630816068
pojazd porusza się w PRAWO
położenie: -0.6422104670672758 prędkość: 0.017417211005663874
pojazd porusza się w PRAWO
położenie: -0.6239223229402759 prędkość: 0.01828814412699989
pojazd porusza się w PRAWO
położenie: -0.6048930604206525 prędkość: 0.01902926251962342
pojazd porusza się w PRAWO
położenie: -0.5852601169283335 prędkość: 0.01963294349231898
pojazd porusza się w PRAWO
położenie: -0.5651673463607513 prędkość: 0.02009277056758223
pojazd porusza się w PRAWO
położenie: -0.5447636189521472 prędkość: 0.020403727408604033
pojazd porusza się w PRAWO
położenie: -0.5242012618559381 prędkość: 0.020562357096209165
pojazd porusza się w PRAWO
położenie: -0.5036343861152569 prędkość: 0.020566875740681126

```

```

pojazd porusza się w PRAWO
położenie: -0.48321715379164004   prędkość:  0.02041723232361688
pojazd porusza się w PRAWO
położenie: -0.46310204337017835   prędkość:  0.02011511042146171
pojazd porusza się w PRAWO
położenie: -0.4434381716854864   prędkość:  0.019663871684691952
pojazd porusza się w PRAWO
położenie: -0.4243697265029366   prędkość:  0.019068445182549815
pojazd porusza się w PRAWO
położenie: -0.4060345559844586   prędkość:  0.018335170518477967
pojazd porusza się w PRAWO
położenie: -0.3885629503948938   prędkość:  0.017471605589564804
pojazd porusza się w PRAWO
położenie: -0.37207663865329377   prędkość:  0.016486311741600007
pojazd porusza się w PRAWO
położenie: -0.35668800890234265   prędkość:  0.015388629750951134
pojazd porusza się w PRAWO
położenie: -0.3424995493108534   prędkość:  0.014188459591489221
pojazd porusza się w PRAWO
położenie: -0.3296034938163245   prędkość:  0.012896055494528953
pojazd porusza się w PRAWO
położenie: -0.3180816481732261   prędkość:  0.011521845643098362
pojazd porusza się w PRAWO
położenie: -0.3080053649184215   prędkość:  0.010076283254804568
pojazd porusza się w PRAWO
położenie: -0.29943563183387684   prędkość:  0.008569733084544676
pojazd porusza się w PRAWO
położenie: -0.2924232370768251   prędkość:  0.007012394757051751
pojazd porusza się w PRAWO
położenie: -0.28700897508596507   prędkość:  0.005414261990860022
pojazd porusza się w PRAWO

```

Po zmianach położenia widać, że **pojazd początkowo kontynuuje ruch w prawo** (wartość położenia ulega zwiększeniu) czyli **podjeżdża pod wzniesienie z prawej strony**. Po pewnym czasie jednak **prędkość zmienia wartość z dodatniej na ujemną** co oznacza, że **pojazd zaczyna opadać czyli porusza się w lewo**.

## ▼ Nagrody

Nagroda wynosi **-1 za każdy krok czasowy**, aż do osiągnięcia **docelowej pozycji 0.5** (położenie chorągiewki).

## ▼ Zakończenie epizodu

Zakończenie epizodu następuje w chwili **osiągnięcia położenia 0.5** lub **po przejściu 200 kroków**.

Przetestujmy działanie środowiska w przypadku **losowego wyboru akcji** (action =

```
R = 0
env.reset()
for e in range(300):
    #env.render()
    action = env.action_space.sample()
    obs, reward, done, info = env.step(action)
    print("Epizod=",e," akcja=",action," R=",reward," koniec epizodu?=",done," stan=",obs)
```

Epizod= 176	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.50715639 0.00311]
Epizod= 177	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.50309155 0.0040]
Epizod= 178	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-0.49918041 0.0039]
Epizod= 179	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.49445225 0.0047]
Epizod= 180	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.4909424 0.0035]
Epizod= 181	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.48667709 0.0042]
Epizod= 182	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.48368813 0.0029]
Epizod= 183	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.48199777 0.0016]
Epizod= 184	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.47961862 0.0023]
Epizod= 185	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.47656837 0.0030]
Epizod= 186	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-0.47386967 0.0026]
Epizod= 187	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.47254256 0.0013]
Epizod= 188	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-4.72596870e-01 -5.4]
Epizod= 189	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-4.73032208e-01 -4.]
Epizod= 190	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.47484534 -0.0018]
Epizod= 191	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.47602282 -0.0011]
Epizod= 192	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-0.47755592 -0.0015]
Epizod= 193	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-0.47943323 -0.0018]
Epizod= 194	akcja= 1	R= -1.0	koniec epizodu?= False	stan= [-0.48164082 -0.0022]
Epizod= 195	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.48516227 -0.0035]
Epizod= 196	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.48797135 -0.0028]
Epizod= 197	akcja= 0	R= -1.0	koniec epizodu?= False	stan= [-0.49204712 -0.0040]
Epizod= 198	akcja= 2	R= -1.0	koniec epizodu?= False	stan= [-0.49535918 -0.0033]
Epizod= 199	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.49888279 -0.00352]
Epizod= 200	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-0.50359159 -0.00470]
Epizod= 201	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.50845036 -0.00485]
Epizod= 202	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.5124227 -0.00397]
Epizod= 203	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.51647885 -0.00405]
Epizod= 204	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.52058839 -0.00410]
Epizod= 205	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.52472051 -0.00413]
Epizod= 206	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-0.52984422 -0.00512]
Epizod= 207	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.53492109 -0.00507]
Epizod= 208	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.53991306 -0.00499]
Epizod= 209	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.54478272 -0.00486]
Epizod= 210	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.5484936 -0.00371]
Epizod= 211	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-0.55301795 -0.00452]
Epizod= 212	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.55632195 -0.00330]
Epizod= 213	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.55838091 -0.00205]
Epizod= 214	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.55917948 -0.00079]
Epizod= 215	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-5.59711698e-01 -5.3]
Epizod= 216	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-5.59973601e-01 -2.6]
Epizod= 217	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.55896323 0.00101]
Epizod= 218	akcja= 2	R= -1.0	koniec epizodu?= True	stan= [-0.55668813 0.00227]
Epizod= 219	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.55416527 0.00252]
Epizod= 220	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-0.55241347 0.00175]
Epizod= 221	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.55044584 0.00196]
Epizod= 222	akcja= 1	R= -1.0	koniec epizodu?= True	stan= [-0.54827707 0.00216]
Epizod= 223	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-0.54692338 0.00135]
Epizod= 224	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-5.46394899e-01 5.2]
Epizod= 225	akcja= 0	R= -1.0	koniec epizodu?= True	stan= [-5.46695582e-01 -3.0]

```
Epizod= 226 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-5.46823177e-01 -1.2
Epizod= 227 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.54577673 0.00104
Epizod= 228 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.54356407 0.00221
Epizod= 229 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.54220176 0.00136
Epizod= 230 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-5.41700002e-01 5.0
Epizod= 231 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.54006255 0.00163
Epizod= 232 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.53730167 0.00276
```

Jak widać po 200 krokach następuje zakończenie epizodu.

## ➤ Rozwiązanie zadania 1 z zestawu 11

```
#DO UZUPEŁNIENIA
```

```
env.reset()
```

```
for _ in range(80):
```

```
    action = 0
```

```
    obs, reward, done, info = env.step(action)
```

```
    print("położenie: ",obs[0]," prędkość: ",obs[1])
```

```
    if obs[1]<0:
```

```
        print("pojazd porusza się w LEWO")
```

```
    else:
```

```
        print("pojazd porusza się w PRAWO")
```

```
        action = 2
```

```
położenie: -0.4838008641102771 prędkość: -0.0013075098966793728
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.4864061497061203 prędkość: -0.002605285595843153
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.49028980156363206 prędkość: -0.003883651857511746
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.49542285518006735 prędkość: -0.005133053616435269
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5017669766453619 prędkość: -0.006344121465294497
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.50927471956452 prędkość: -0.007507742919158122
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5178898598433522 prędkość: -0.00861514027883229
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5275478148971643 prędkość: -0.009657955053812079
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5381761528490604 prędkość: -0.010628337951896053
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5496951953172741 prędkość: -0.011519042468213659
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5620187145143152 prędkość: -0.012323519197041093
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5750547217485559 prędkość: -0.013036007234240652
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.5887063402688328 prędkość: -0.013651618520276972
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.6028727510314188 prędkość: -0.014166410762585985
```

```
pojazd porusza się w LEWO
```

```
położenie: -0.6174501957475762 prędkość: -0.014577444716157382
```

```

pojazd porusza się w LEWO
położenie: -0.6323330178521861 prędkość: -0.014882822104609844
pojazd porusza się w LEWO
położenie: -0.6474147191579878 prędkość: -0.015081701305801642
pojazd porusza się w LEWO
położenie: -0.6625890082022501 prędkość: -0.015174289044262327
pojazd porusza się w LEWO
położenie: -0.6777508158364507 prędkość: -0.015161807634200566
pojazd porusza się w LEWO
położenie: -0.6927972545236789 prędkość: -0.015046438687228181
pojazd porusza się w LEWO
położenie: -0.7076285000402646 prędkość: -0.014831245516585611
pojazd porusza się w LEWO
położenie: -0.722148577660714 prędkość: -0.014520077620449457
pojazd porusza się w LEWO
położenie: -0.7362660391785915 prędkość: -0.0141174615178774
pojazd porusza się w LEWO
położenie: -0.7498945219589531 prędkość: -0.013628482780361708
pojazd porusza się w LEWO
położenie: -0.7629531862833964 prędkość: -0.013058664324443296
pojazd porusza się w LEWO
położenie: -0.775367032200474 prędkość: -0.012413845917077638
pojazd porusza się w LEWO
położenie: -0.7870671016375392 prędkość: -0.01170006943706529
pojazd porusza się w LEWO
położenie: -0.7979905754343312 prędkość: -0.01092347379679188
pojazd porusza się w LEWO
położenie: -0.8080807780690117 prędkość: -0.010090202634680568
pojazd porusza się w LEWO

```

## Rozwiązanie zadania 2 z zestawu 11

```

env = gym.make('MountainCar-v0')
env.reset()

array([-0.56321908, 0.          ])

#https://gist.github.com/gkhayes/3d154e0505e31d6367be22ed3da2e955

def QLearning(env, learning, discount, epsilon, min_eps, episodes):

    num_states = (env.observation_space.high - env.observation_space.low)*\
        np.array([10, 100])
    num_states = np.round(num_states, 0).astype(int) + 1

    Q = np.random.uniform(low = -1, high = 1, size = (num_states[0], num_states[1], env.acti

    reward_list = []
    ave_reward_list = []

    for i in range(episodes):

        epsilon = epsilon - (1/episodes)
        done = False
        tot_reward, reward = 0,0
        state = env.reset()

```



```

state_adj = (state - env.observation_space.low)*np.array([10, 100])
state_adj = np.round(state_adj, 0).astype(int)

while done != True:

    if i >= (episodes - 20):
        env.render()

    if np.random.random() < 1 - epsilon:
        action = np.argmax(Q[state_adj[0], state_adj[1]])
    else:
        action = np.random.randint(0, env.action_space.n)

    state2, reward, done, info = env.step(action)

    state2_adj = (state2 - env.observation_space.low)*np.array([10, 100])
    state2_adj = np.round(state2_adj, 0).astype(int)

    if done and state2[0] >= 0.5:
        Q[state_adj[0], state_adj[1], action] = reward

    else:
        delta = learning*(reward +discount*np.max(Q[state2_adj[0],state2_adj[1]])
        Q[state_adj[0], state_adj[1],action] += delta

    tot_reward += reward
    state_adj = state2_adj

    reward_list.append(tot_reward)

    if (i+1) % 100 == 0:
        ave_reward = np.mean(reward_list)
        ave_reward_list.append(ave_reward)
        reward_list = []

    if (i+1) % 100 == 0:
        print('Episode {} Average Reward: {}'.format(i+1, ave_reward))

env.close()

return ave_reward_list

```

Te rzeczy ponizej są bo mialem jakieś problemy z collabem

```
pip install pygame==1.5.11
```

```
%%bash
```

```
# install required system dependencies
```

```
apt-get install -y xvfb x11-utils
```

```
# install required python dependencies (might need to install additional gym extras depend  
pip install gym[box2d]==0.17.* pyvirtualdisplay==0.2.* PyOpenGL==3.1.* PyOpenGL-accelerate
```

```
#import
```

```
import pyvirtualdisplay
```

```
_display = pyvirtualdisplay.Display(visible=0, # remember to use visible=0 and not False  
                                     size=(1400, 900))
```

```
_ = _display.start()
```

```
#check
```

```
!echo $DISPLAY
```

```
# Run Q-learning algorithm
```

```
rewards = QLearning(env, 0.2, 0.9, 0.8, 0, 5000)
```

```

Episode 100 Average Reward: -200.0
Episode 200 Average Reward: -200.0
Episode 300 Average Reward: -200.0
Episode 400 Average Reward: -200.0
Episode 500 Average Reward: -200.0
Episode 600 Average Reward: -200.0
Episode 700 Average Reward: -200.0
Episode 800 Average Reward: -200.0
Episode 900 Average Reward: -200.0
Episode 1000 Average Reward: -200.0
Episode 1100 Average Reward: -200.0
Episode 1200 Average Reward: -200.0
Episode 1300 Average Reward: -200.0
Episode 1400 Average Reward: -200.0
Episode 1500 Average Reward: -200.0
Episode 1600 Average Reward: -200.0
Episode 1700 Average Reward: -200.0
Episode 1800 Average Reward: -200.0
Episode 1900 Average Reward: -200.0
Episode 2000 Average Reward: -200.0
Episode 2100 Average Reward: -200.0
Episode 2200 Average Reward: -200.0
Episode 2300 Average Reward: -200.0
Episode 2400 Average Reward: -200.0
Episode 2500 Average Reward: -200.0
Episode 2600 Average Reward: -199.56
Episode 2700 Average Reward: -199.93
Episode 2800 Average Reward: -200.0
Episode 2900 Average Reward: -199.96
Episode 3000 Average Reward: -198.33
Episode 3100 Average Reward: -195.81
Episode 3200 Average Reward: -196.96
Episode 3300 Average Reward: -197.63
Episode 3400 Average Reward: -198.8
Episode 3500 Average Reward: -200.0
Episode 3600 Average Reward: -197.47

```

```

plt.plot(100*(np.arange(len(rewards)) + 1), rewards)
plt.xlabel('Episodes')
plt.ylabel('Average Reward')
plt.title('Average Reward vs Episodes')
plt.savefig('rewards.jpg')
plt.close()

```

```

-----
NameError                                Traceback (most recent call last)
<ipython-input-11-15127dbacbe8> in <module>()
----> 1 plt.plot(100*(np.arange(len(rewards)) + 1), rewards)
      2 plt.xlabel('Episodes')
      3 plt.ylabel('Average Reward')
      4 plt.title('Average Reward vs Episodes')
      5 plt.savefig('rewards.jpg')

```

**NameError:** name 'rewards' is not defined

SEARCH STACK OVERFLOW

