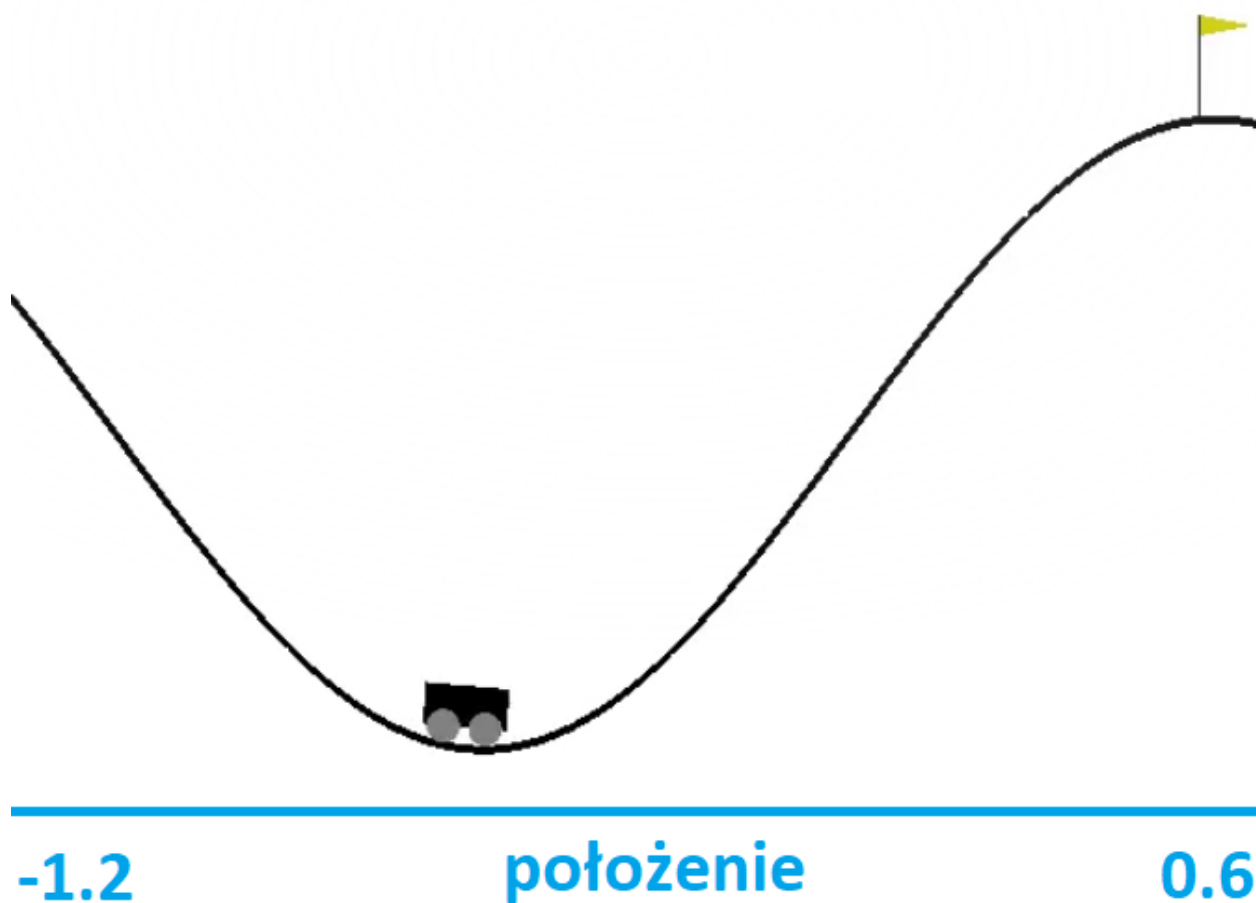


▼ Środowisko MountainCar

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 iz powrotem", w lewo i prawo i nabranie rozpędu.**



```
import numpy as np
import gym
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
from keras.models import Sequential
from keras.layers import Dense
```

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([-0.50000892,  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
```

```
model = Sequential()
model.add(Dense(units = 20, input_dim=2, activation='relu'))
model.add(Dense(units = 20, activation = "relu"))
model.add(Dense(units = 3, activation = "linear"))
```

```
opt = tf.keras.optimizers.Adam(learning_rate=0.001)
#opt = keras.optimizers.SGD(learning_rate=0.001)
```

```
model.compile(loss='MSE',optimizer=opt)
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 20)	60
dense_1 (Dense)	(None, 20)	420
dense_2 (Dense)	(None, 3)	63
Total params: 543		
Trainable params: 543		
Non-trainable params: 0		

▼ 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= [-0.47345975 -0.00138489]
    stan= [-0.47621927 -0.00275952]
    stan= [-0.48033294 -0.00411367]
```

```

stan= [-0.48577019 -0.00543725]
stan= [-0.49249055 -0.00672036]
stan= [-0.50044388 -0.00795333]
stan= [-0.50957074 -0.00912685]
stan= [-0.51980277 -0.01023203]
stan= [-0.53106327 -0.0112605 ]
stan= [-0.5432678  -0.01220452]

```

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 pojazdu w lewą stronę):

```

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:  -0.7209211707043290  prędkość:  -0.014243443037009237
    pojazd porusza się w LEWO
    położenie:  -0.7347775908799211  prędkość:  -0.013850420095391239
    pojazd porusza się w LEWO
    położenie:  -0.7481480144416632  prędkość:  -0.013370423561742157
    pojazd porusza się w LEWO
    położenie:  -0.7609588354819579  prędkość:  -0.012810821040294596
    pojazd porusza się w LEWO
    położenie:  -0.7731361318009113  prędkość:  -0.012177296318953444
    pojazd porusza się w LEWO
    położenie:  -0.7846118716952136  prędkość:  -0.011475739894302301
    pojazd porusza się w LEWO
    położenie:  -0.7953240194923256  prędkość:  -0.010712147797111883
    pojazd porusza się w LEWO
    położenie:  -0.8052165519990157  prędkość:  -0.009892532506690074

```

położenie: -0.8092189319990137 prędkość: -0.009022848121356848
pojazd porusza się w LEWO
położenie: -0.8142394001203725 prędkość: -0.009022848121356848
pojazd porusza się w LEWO
położenie: -0.8223483312422367 prędkość: -0.008108931121864192
pojazd porusza się w LEWO
położenie: -0.8295047885206265 prędkość: -0.007156457278389774
pojazd porusza się w LEWO
położenie: -0.8356757030699199 prędkość: -0.006170914549293361
pojazd porusza się w LEWO
położenie: -0.8408332942947363 prędkość: -0.005157591224816395
pojazd porusza się w LEWO
położenie: -0.8449548723851239 prędkość: -0.004121578090387549
pojazd porusza się w LEWO
położenie: -0.8480226554042847 prędkość: -0.0030677830191607925
pojazd porusza się w LEWO
położenie: -0.8500236115443487 prędkość: -0.002000956140064006
pojazd porusza się w LEWO
położenie: -0.8509493350932271 prędkość: -0.0009257235488783788
pojazd porusza się w LEWO
położenie: -0.8507959625097952 prędkość: 0.00015337258343195133
pojazd porusza się w PRAWO
położenie: -0.8495641327926646 prędkość: 0.0012318297171305189
pojazd porusza się w PRAWO
położenie: -0.8472589940813146 prędkość: 0.0023051387113500327
pojazd porusza się w PRAWO
położenie: -0.8438902561682768 prędkość: 0.003368737913037658
pojazd porusza się w PRAWO
położenie: -0.839472286343145 prędkość: 0.004417969825131873
pojazd porusza się w PRAWO
położenie: -0.8340242437499873 prędkość: 0.005448042593157677
pojazd porusza się w PRAWO
położenie: -0.8275702452427494 prędkość: 0.0064539985072379295
pojazd porusza się w PRAWO
położenie: -0.8201395536040917 prędkość: 0.007430691638657661
pojazd porusza się w PRAWO
położenie: -0.8117667770039253 prędkość: 0.008372776600166398
pojazd porusza się w PRAWO
położenie: -0.8024920667856009 prędkość: 0.009274710218324402
pojazd porusza się w PRAWO
położenie: -0.7923612991700723 prędkość: 0.010130767615528696
pojazd porusza się w PRAWO
położenie: -0.7814262253671163 prędkość: 0.010935073802956021
pojazd porusza się w PRAWO

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:  -0.31795163957143770  prędkość:  0.016217655405559052
    pojazd porusza się w PRAWO
    położenie:  -0.30116026432247023  prędkość:  0.01677137524898751
    pojazd porusza się w PRAWO
    położenie:  -0.28593608809671744  prędkość:  0.015224176225752761
    pojazd porusza się w PRAWO
    położenie:  -0.27234715409692395  prędkość:  0.013588933999793511
    pojazd porusza się w PRAWO
    położenie:  -0.26046917344943016  prędkość:  0.011877980647493791
    pojazd porusza się w PRAWO
    położenie:  -0.2503660001870513  prędkość:  0.010103173262378879
    pojazd porusza się w PRAWO
    położenie:  -0.2420901768952206  prędkość:  0.008275823291830683
    pojazd porusza się w PRAWO
    położenie:  -0.23568349431372076  prędkość:  0.006406682581499837
    pojazd porusza się w PRAWO
    położenie:  -0.23117751465544747  prędkość:  0.004505979658273278
    pojazd porusza się w PRAWO
    położenie:  -0.22859401652706965  prędkość:  0.0025834981283778353
    pojazd porusza się w PRAWO
    położenie:  -0.22794532847034368  prędkość:  0.0006486880567259767
    pojazd porusza się w PRAWO
    położenie:  -0.22923452782086612  prędkość:  -0.001289199350522424
    pojazd porusza się w LEWO
    położenie:  -0.23245549150796624  prędkość:  -0.003220963687100122
```

pojazd porusza się w LEWO

położenie: -0.2375927954369454 prędkość: -0.0051373039289791455

pojazd porusza się w LEWO

położenie: -0.2446214691290125 prędkość: -0.007028673692067118

pojazd porusza się w LEWO

położenie: -0.2535066222929232 prędkość: -0.008885153163910718

pojazd porusza się w LEWO

położenie: -0.2642029698702539 prędkość: -0.0106963475773307

pojazd porusza się w LEWO

położenie: -0.2766542916349516 prędkość: -0.01245132176469771

pojazd porusza się w LEWO

położenie: -0.29079287128809567 prędkość: -0.014138579653144065

pojazd porusza się w LEWO

położenie: -0.3065389676447339 prędkość: -0.01574609635663819

pojazd porusza się w LEWO

położenie: -0.3238003762634373 prędkość: -0.017261408618703407

pojazd porusza się w LEWO

położenie: -0.3424721428959649 prędkość: -0.018671766632527616

pojazd porusza się w LEWO

położenie: -0.36243648957202845 prędkość: -0.019964346676063543

pojazd porusza się w LEWO

położenie: -0.38356300921682324 prędkość: -0.0211265196447948

pojazd porusza się w LEWO

położenie: -0.4057091749225535 prędkość: -0.022146165705730223

pojazd porusza się w LEWO

położenie: -0.4287211952895813 prędkość: -0.023012020367027798

pojazd porusza się w LEWO

położenie: -0.45243522812633985 prędkość: -0.02371403283675856

pojazd porusza się w LEWO

położenie: -0.4766789424108754 prędkość: -0.024243714284535558

pojazd porusza się w LEWO

położenie: -0.5012733945527624 prędkość: -0.024594452141887095

pojazd porusza się w LEWO

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 = env.action_space.sample()`).

```
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= 242 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.41901747 -0.01009755]
    Epizod= 243 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.43068065 -0.01086318]
    Epizod= 244 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.44323172 -0.01255107]
    Epizod= 245 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.45737972 -0.014148 ]
    Epizod= 246 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.47102111 -0.01364139]
    Epizod= 247 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.48505519 -0.01403408]
    Epizod= 248 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.50037771 -0.01532252]
    Epizod= 249 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.51587425 -0.01549653]
    Epizod= 250 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.53242871 -0.01655446]
    Epizod= 251 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.54891696 -0.01648825]
    Epizod= 252 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.5642155 -0.01529854]
    Epizod= 253 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.57821017 -0.01399467]
    Epizod= 254 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.59079708 -0.01258692]
    Epizod= 255 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.60388342 -0.01308633]
    Epizod= 256 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.61737342 -0.01349 ]
    Epizod= 257 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.63016936 -0.01279594]
    Epizod= 258 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.64317956 -0.0130102 ]
    Epizod= 259 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.65631202 -0.01313246]
    Epizod= 260 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.66847522 -0.0121632 ]
    Epizod= 261 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.68058574 -0.01211051]
    Epizod= 262 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.69056188 -0.00997616]
```



```

Epizod= 262 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.69038187 -0.00337818]
Epizod= 263 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.69933754 -0.00877564]
Epizod= 264 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.70685536 -0.00751782]
Epizod= 265 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.71406695 -0.00721159]
Epizod= 266 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.7199265 -0.00585956]
Epizod= 267 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.72439727 -0.00447077]
Epizod= 268 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.72845149 -0.00405422]
Epizod= 269 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.73006422 -0.00161273]
Epizod= 270 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-7.30225590e-01 -1.61368917e-04]
Epizod= 271 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-7.29934614e-01 2.90975710e-04]
Epizod= 272 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.72719307 0.00274154]
Epizod= 273 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.72401775 0.00317532]
Epizod= 274 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.71942824 0.00458951]
Epizod= 275 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.71245304 0.0069752 ]
Epizod= 276 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.70313601 0.00931703]
Epizod= 277 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.69153662 0.01159939]
Epizod= 278 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.6787303 0.01280632]
Epizod= 279 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.66580205 0.01292826]
Epizod= 280 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.65283932 0.01296272]
Epizod= 281 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.6399314 0.01290792]
Epizod= 282 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.62616859 0.01376281]

Epizod= 283 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.61264858 0.01352 ]
Epizod= 284 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.59846862 0.01417996]
Epizod= 285 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.58473184 0.01373678]
Epizod= 286 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.57153914 0.01319271]
Epizod= 287 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.55798811 0.01355102]
Epizod= 288 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.54417963 0.01380849]
Epizod= 289 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.53121688 0.01296274]
Epizod= 290 akcja= 0 R= -1.0 koniec epizodu?= True stan= [-0.51919701 0.01201987]
Epizod= 291 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.50721015 0.01198686]
Epizod= 292 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.49534615 0.011864 ]
Epizod= 293 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.48269379 0.01265236]
Epizod= 294 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.47034745 0.01234634]
Epizod= 295 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.4573988 0.01294865]
Epizod= 296 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.4439434 0.01345541]
Epizod= 297 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.43007973 0.01386366]
Epizod= 298 akcja= 2 R= -1.0 koniec epizodu?= True stan= [-0.4159083 0.01417143]
Epizod= 299 akcja= 1 R= -1.0 koniec epizodu?= True stan= [-0.40253057 0.01337773]

```

Jak widać po 200 krokach następuje zakończenie epizodu.

▼ Rozwiązanie zadania 1 z zestawu 11

```
train_episodes = 100
epsilon = 1
gamma = 0.99
max_steps = 50
max_right = -1.2

Loss = []
Rewards = []

for e in range(1, train_episodes+1):

    epsilon = epsilon-(1/train_episodes)
    max_right = -1.2
    total_reward = 0
    t = 0

    state = env.reset()
    state = np.reshape(state, [1, 2])

    done = False
    while t < max_steps and done == False:

        Qs = model.predict(state)[0]

        if np.random.rand()<epsilon:
            action = env.action_space.sample()
        else:
            action = np.argmax(Qs)

        next_state, reward, done, _ = env.step(action)
        next_state = np.reshape(next_state, [1, 2])
        total_reward += reward

        if max_right<next_state[0][0]:
            max_right = next_state[0][0]
```

```

if done:
    y = reward
else:
    y = reward + gamma*np.max(model.predict(next_state)[0])
Q_target = model.predict(state)
Q_target[0][action] = y

h = model.fit(state,Q_target,epochs=1,verbose=0)

loss = h.history['loss'][0]

state = next_state
t=t+1

print(e," Max right=",max_right," Loss=",loss)
Rewards.append(max_right)
Loss.append(loss)
43 Max right= -0.4544398150445523 Loss= 0.03039950230717059
44 Max right= -0.39982048851691937 Loss= 0.02115243673324585
45 Max right= -0.3250188571889409 Loss= 0.13363198935985565
46 Max right= -0.4583228308725831 Loss= 0.01023006346076727
47 Max right= -0.34295909083477477 Loss= 0.43051907420158386
48 Max right= -0.45875983285061167 Loss= 0.03335166350007057
49 Max right= -0.303294654076809 Loss= 0.282188355922699
50 Max right= -0.4602290823015457 Loss= 9.203958325088024e-05
51 Max right= -0.4329772571695982 Loss= 0.0008303858921863139
52 Max right= -0.29572695151806305 Loss= 0.06419655680656433
53 Max right= -0.4013258958169693 Loss= 0.10809993743896484
54 Max right= -0.4161369589853836 Loss= 0.22820544242858887
55 Max right= -0.4055272479267484 Loss= 0.16885365545749664
56 Max right= -0.43468814114537235 Loss= 0.06700333207845688
57 Max right= -0.3854312521869322 Loss= 0.045003872364759445
58 Max right= -0.40048516222000874 Loss= 0.344234436750412
59 Max right= -0.31680099861664224 Loss= 0.2707521617412567
60 Max right= -0.458054443074052 Loss= 0.025881638750433922
61 Max right= -0.41240338105002955 Loss= 0.006520465016365051
62 Max right= -0.2623584694308847 Loss= 0.25848543643951416
63 Max right= -0.3988424098750598 Loss= 0.3133184611797333
64 Max right= -0.3179537859772991 Loss= 0.3130522072315216
65 Max right= -0.4723764368957678 Loss= 0.029621867462992668
66 Max right= -0.4198071144890853 Loss= 0.40633103251457214
67 Max right= -0.37000455502214005 Loss= 0.1201000060050000

```

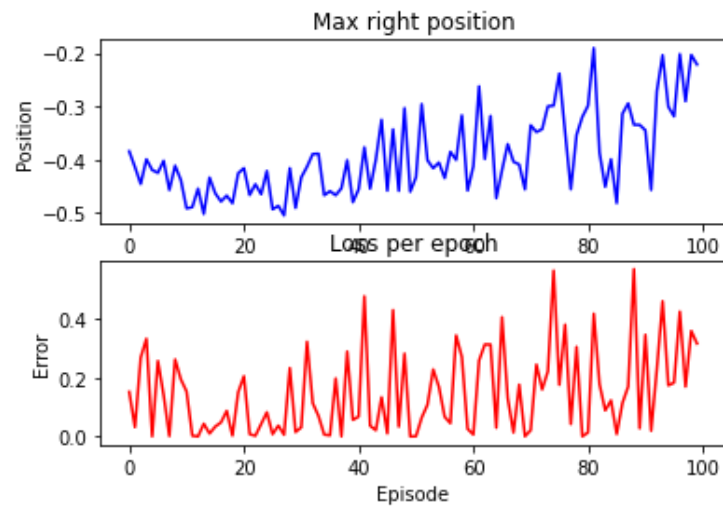
```
67 Max right= -0.37088455503214995 Loss= 0.1301829069852829
68 Max right= -0.4039835462973788 Loss= 0.01335836946964264
69 Max right= -0.4091229256119889 Loss= 0.17648209631443024
70 Max right= -0.4558749884072542 Loss= 0.0005266648367978632
71 Max right= -0.33551313855009135 Loss= 0.02121269889175892
72 Max right= -0.3481310050265987 Loss= 0.24469123780727386
73 Max right= -0.34279031660711434 Loss= 0.15998469293117523
74 Max right= -0.29991529377845366 Loss= 0.2219032198190689
75 Max right= -0.2985213112638564 Loss= 0.5649173855781555
76 Max right= -0.2383904639370511 Loss= 0.17550456523895264
77 Max right= -0.34824776734845975 Loss= 0.3807736933231354
78 Max right= -0.4556516711424482 Loss= 0.04201093316078186
79 Max right= -0.3535971823647202 Loss= 0.3052104413509369
80 Max right= -0.31983792806303646 Loss= 0.00025524309603497386
81 Max right= -0.29878608892407743 Loss= 0.01346754003316164
82 Max right= -0.19056222810321774 Loss= 0.41812968254089355
83 Max right= -0.38664788811031653 Loss= 0.17839322984218597
84 Max right= -0.4509692058589826 Loss= 0.08874634653329849
85 Max right= -0.39924307029886774 Loss= 0.12297166138887405
86 Max right= -0.4816241950215786 Loss= 0.008111022412776947
87 Max right= -0.31381794261239615 Loss= 0.11404786258935928
88 Max right= -0.29434529578625407 Loss= 0.17012667655944824
89 Max right= -0.3345606104995335 Loss= 0.5701470971107483
90 Max right= -0.33482988659262647 Loss= 0.02752547711133957
91 Max right= -0.3449349051299161 Loss= 0.3462531864643097
92 Max right= -0.45693945656631546 Loss= 0.019402828067541122
93 Max right= -0.27147022173411384 Loss= 0.22931361198425293
94 Max right= -0.2040297793721299 Loss= 0.4604019820690155
95 Max right= -0.3006287367408176 Loss= 0.1738826036453247
96 Max right= -0.31886025784422484 Loss= 0.18301813304424286
97 Max right= -0.20195036802811506 Loss= 0.4252749979496002
98 Max right= -0.29044797130380745 Loss= 0.16955304145812988
99 Max right= -0.2035235896866738 Loss= 0.35922861099243164
100 Max right= -0.22145335930646506 Loss= 0.31623461842536926
```

```
plt.subplot(211)
plt.ylabel('Position')
plt.title('Max right position')
plt.plot(list(range(train_episodes)), Rewards, "b")
```

```
plt.subplot(212)
plt.xlabel('Episode')
plt.ylabel('Error')
```

```
plt.title('Loss per epoch')
plt.plot(list(range(train_episodes)), Loss, "r")

plt.show()
```



#DO UZUPEŁNIENIA

✓ 0 s ukończono o 22:52

