First, we import the needed libraries:

```
import pandas as pd
import numpy as np
import nashpy as nash
```

Then, we initialize values for parameters

```
parameters

# cu == Customs Administration
# P == domestic manufacturer

D21 = 300
D22 = 260
D11 = 150
D12 = 100

C1 = 400
C2 = 600

B = 60  # D21 - D22 < D11 - D12 < B
a = 100
s = 250  # s < C1
```

Now we are forming the utility functions for two players. Please note that "Cu" refers to the customs (customs authority) and "P" refers to the car manufacturer.

```
payoffs

cu11 = C1 + a - s ; P11 = D11 - B
cu12 = C2 ; P12 = D12
cu21 = a ; P21 = D21 - B
cu22 = 0 ; P22 = D22
```

Now, we create a simultaneous game using the `nash.Game()` command and obtain Nash equilibria using the support enumeration method.

nash equilibrium

```
cu = np.array([[cu11, cu12], [cu21, cu22]])  #player cu
P = np.array([[P11, P12], [P21, P22]])       #player P

game = nash.Game(cu,P)
game
```

✓ 0.0s

```
Bi matrix game with payoff matrices:

Row player:
[[250 600]
 [100   0]]

Column player:
[[ 90 100]
 [240 260]]
```

```
eqs = game.support_enumeration()    #(import, Consistent quality) is eq
for eq in eqs:
    print(eq)
```

✓ 0.0s

```
(array([1., 0.]), array([0., 1.]))
```

As you can see, the game has been created, and it only has a pure Nash equilibrium at (import, Consistent quality). The first component, representing Customs, has a strategy [1,0]), meaning Customs anticipates imports. The second component [0,1] represents the car manufacturer, indicating that the car manufacturer assumes its second strategy (Consistent quality).

One important point here is that s<C1(minimum of s), meaning customs prefers imports, and B has its best response at the upper bound, sticking with the fixed quality.

Now, by keeping the value of "s" constant and decreasing the value of "B" (B < D11 - D12), where the cost of quality improvement decreases, the car manufacturer wants to increase quality. In this scenario, we create a simultaneous game and determine the equilibria.

So, we have:

For lower B:

```python
B = 45  # B < D11 - D12
cu11 = C1 + a - s ; P11 = D11 - B
cu12 = C2 ; P12 = D12
cu21 = a ; P21 = D21 - B
cu22 = 0 ; P22 = D22
cu = np.array([[cu11, cu12], [cu21, cu22]])  #player cu
P = np.array([[P11, P12], [P21, P22]])       #player P
game = nash.Game(cu,P)

eqs = game.support_enumeration()    #(import, increasing quality) is eq
for eq in eqs:
    print(eq)
```

✓ 0.0s

```
(array([1., 0.]), array([1., 0.]))
```

As you can see, in this scenario, the pure Nash equilibrium of the game occurs at the point of (import, quality improvement), which corresponds with our calculations and the real world.

Now, by increasing the value of "s" (shame) above C1 and setting the value of "B" at its lower bound, we know that the Nash equilibrium in this case involves quality improvement while preventing imports.

So, we check with our code, and we have:

For higher s:

```python
B = 25  #B < D21 - D22 < D11 - D12
s = 550  # s > C1
cu11 = C1 + a - s ; P11 = D11 - B
cu12 = C2 ; P12 = D12
cu21 = a ; P21 = D21 - B
cu22 = 0 ; P22 = D22
cu = np.array([[cu11, cu12], [cu21, cu22]])  #player cu
P = np.array([[P11, P12], [P21, P22]])       #player P
game = nash.Game(cu,P)

eqs = game.support_enumeration()    #(Import ban, increasing quality) is eq
for eq in eqs:
    print(eq)
```

✓ 0.0s

```
(array([0., 1.]), array([1., 0.]))
```

In the end, when we execute the mixed Nash equilibria section, we know that because both "p" and "q" are positive numbers, the condition D21 – D22 < B < D11 – D12, s>C1 must hold. Therefore, we set these values accordingly and determine the equilibria.

mixed strategy Nash eq

```
B = 48  # D21 - D22 < B < D11 - D12
s = 600 # s > C1
# C1 = 430

cu11 = C1 + a - s ; P11 = D11 - B
cu12 = C2 ; P12 = D12
cu21 = a ; P21 = D21 - B
cu22 = 0 ; P22 = D22
cu = np.array([[cu11, cu12], [cu21, cu22]])  #player cu
P = np.array([[P11, P12], [P21, P22]])       #player P
game = nash.Game(cu,P)

eqs = game.support_enumeration()
for eq in eqs:
    print(eq)
```

✓ 0.0s

```
(array([0.8, 0.2]), array([0.75, 0.25]))
```

As you can observe, in this range, there is only one mixed Nash equilibrium, where p = 0.8 and q = 0.96. This means that Customs has an 80% inclination towards imports, while the car manufacturer has a 96% inclination towards quality improvement, which aligns with our calculations.