```
import random


# Define the possible cards for Alice and Bob
alice_cards = ["Ace", "King", "Queen", "Jack"]
bob_cards = ["10", "9", "8", "7"]

# Prior probabilities of Alice and Bob having each card
alice_prior = [0.4, 0.3, 0.2, 0.1]
bob_prior = [0.25, 0.25, 0.25, 0.25]


# Define Alice's strategy: a dictionary of actions for each card
alice_strategy = {
    "Ace": "Bet",
    "King": "Bet",
    "Queen": "Check",
    "Jack": "Check"
}

# Define Bob's strategy: a dictionary of actions for each card
bob_strategy = {
    "10": "Bet",
    "9": "Bet",
    "8": "Check",
    "7": "Check"
}


# Simulate the game multiple times to find Bayesian Nash equilibrium
num_simulations = 10000
alice_wins = 0
bob_wins = 0

for _ in range(num_simulations):
    alice_card = random.choices(alice_cards, alice_prior)[0]
    bob_card = random.choices(bob_cards, bob_prior)[0]

    alice_action = alice_strategy[alice_card]
    bob_action = bob_strategy[bob_card]

    if alice_action == "Bet" and bob_action == "Check":
        alice_wins += 1
    elif alice_action == "Check" and bob_action == "Bet":
        bob_wins += 1


# Calculate the probabilities of Alice and Bob winning
alice_probability = alice_wins / num_simulations
bob_probability = bob_wins / num_simulations

print(f"Alice's strategy: {alice_strategy}")
print(f"Bob's strategy: {bob_strategy}")
print(f"Probability of Alice winning: {alice_probability}")
print(f"Probability of Bob winning: {bob_probability}")
```

```
Alice's strategy: {'Ace': 'Bet', 'King': 'Bet', 'Queen': 'Check', 'Jack': 'Check'}
Bob's strategy: {'10': 'Bet', '9': 'Bet', '8': 'Check', '7': 'Check'}
Probability of Alice winning: 0.3542
Probability of Bob winning: 0.1454
```