

Sheet 2

```

from Model import Model
from GameHistoryAnalysis import GameHistoryAnalysis
import matplotlib.pyplot as plt

# Investigation into token number relative to cooperative epochs for 1, 2, 3, 4 Tokens, averaged over
number_of_agents = 50
generations = 1000
game = 'PD'
learning_mechanism = 'GA'
tokens = 4
timeout = 100
iterations = 5
states = 8

cooperative_epochs_by_states_by_tokens = []
avg_percent_outcomes_by_states_by_tokens = []
for s in range(states):
    print('Starting Computation for computational capacity of ' + str(s))
    cooperative_epochs_by_tokens = []
    avg_percent_outcomes_by_tokens = []
    for t in range(tokens):
        print('Starting Computation for communication tokens: ' + str(t))
        total_coop_epochs = 0
        avg_percent_outcomes = [0, 0, 0]
        for i in range(iterations):
            print('Starting Iteration ' + str(i))
            model = Model(number_of_agents, s+1, t+1, generations, learning_mechanism, timeout, game)
            model.run_model()
            gha = GameHistoryAnalysis(model.game_history, number_of_agents)
            total_coop_epochs += gha.cooperative_epochs

            total = sum(gha.outcome_frequency)
            percent_outcomes = [(gha.outcome_frequency[m]/total) * 100 for m in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[j] + percent_outcomes[j] for j in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[k]/iterations for k in range(3)]
            avg_percent_outcomes_by_tokens.append(avg_percent_outcomes)
            cooperative_epochs_by_tokens.append(total_coop_epochs/iterations)
        avg_percent_outcomes_by_states_by_tokens.append(avg_percent_outcomes_by_tokens)
    cooperative_epochs_by_states_by_tokens.append(cooperative_epochs_by_tokens)

```

```

Starting Computation for computational capacity of 0
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3

```

```
Starting Iteration 4
Starting Computation for computational capacity of 1
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 2
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 3
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
```

```
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 4
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 5
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 6
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
```

```

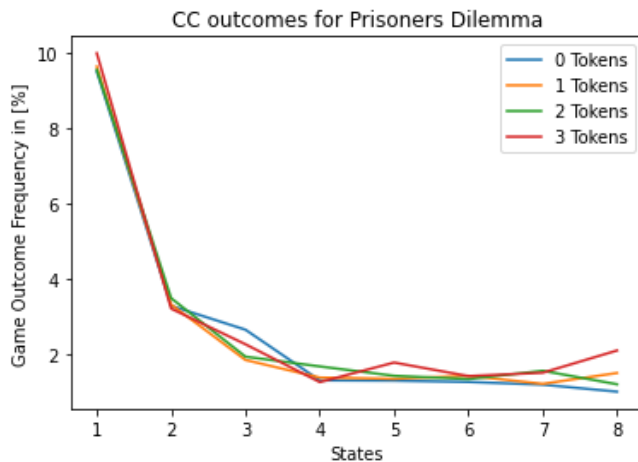
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for computational capacity of 7
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4

```

```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][0] for p in range(players)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('CC outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

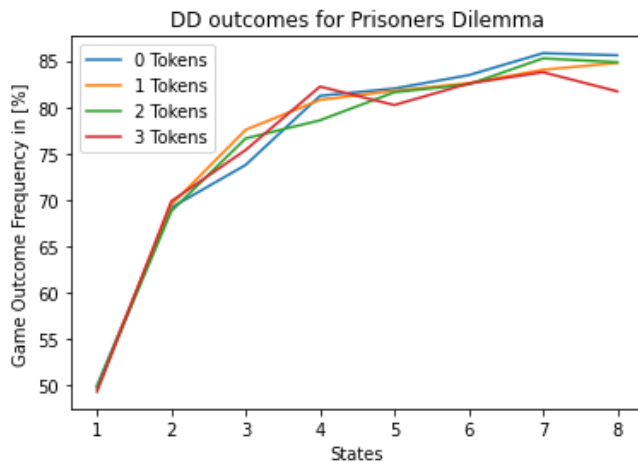
```



```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][1] for p in range(players)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DD outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

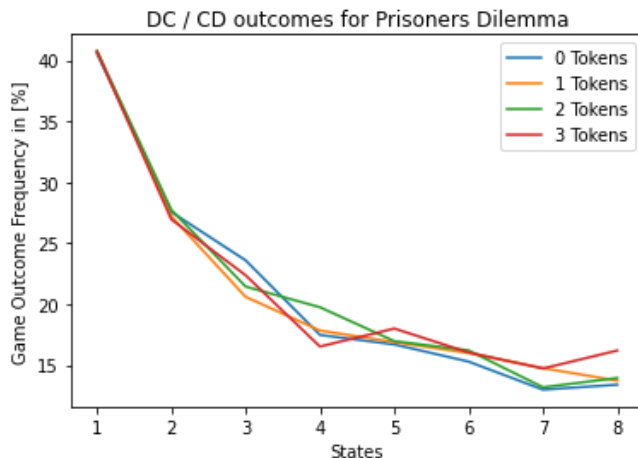
```



```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][2] for p in range(players)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DC / CD outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

```

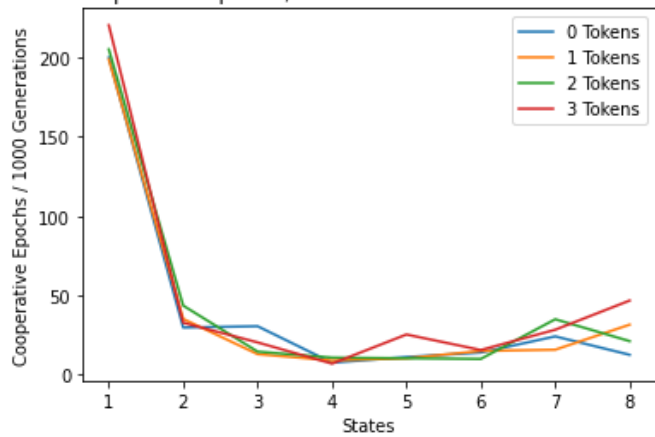


```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [cooperative_epochs_by_states_by_tokens[s][t] for s in range(players)])
plt.ylabel('Cooperative Epochs / 1000 Generations')
plt.xlabel('States')
plt.title('Total Cooperative Epochs / 1000 Generations for Prisoners Dilemma')
plt.legend()
plt.show()

```

Total Cooperative Epochs / 1000 Generations for Prisoners Dilemma

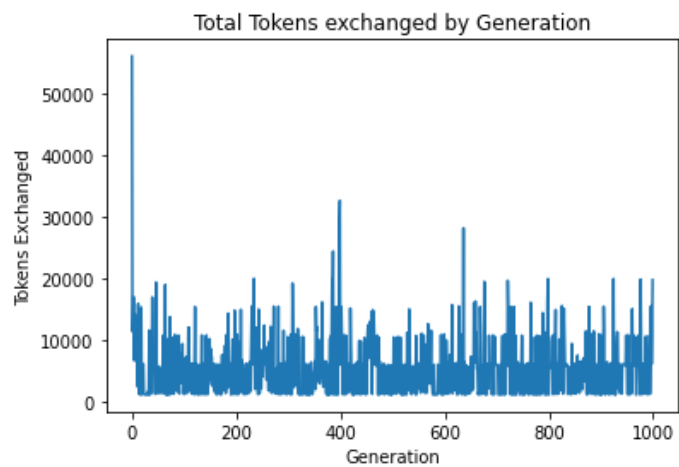
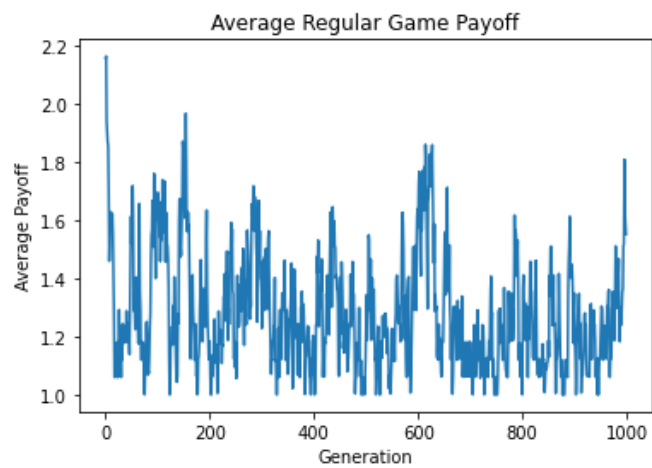
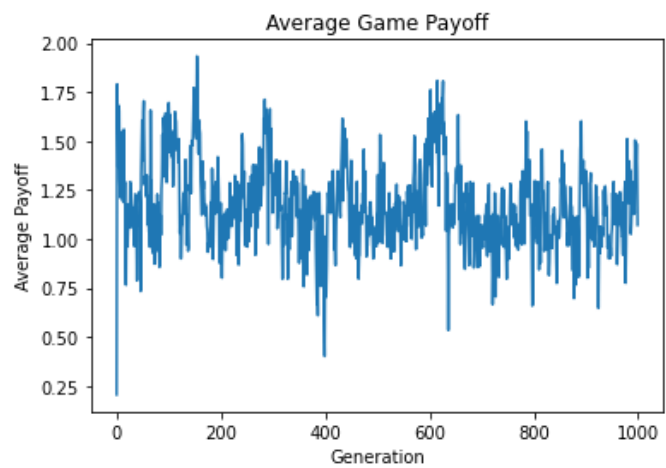


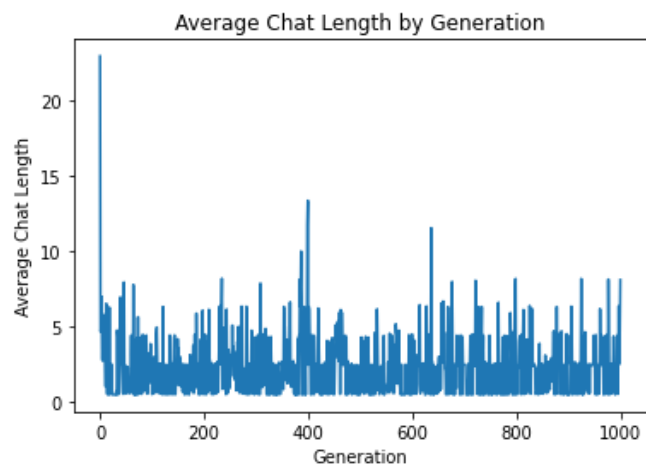
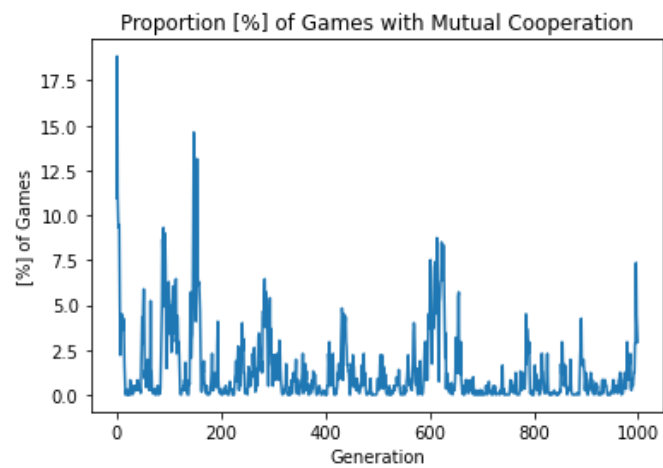
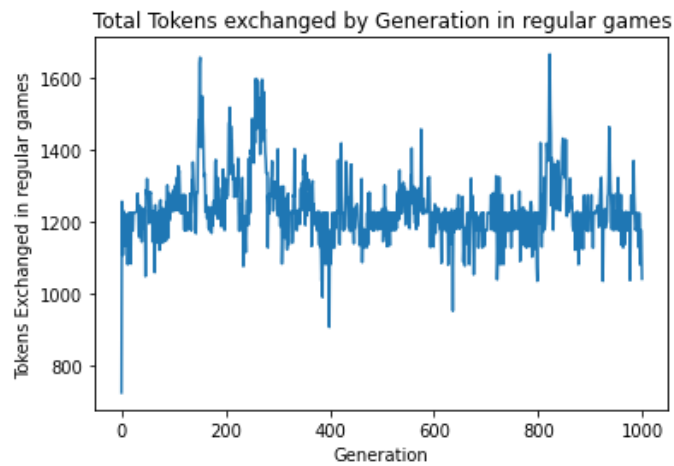
```
print(avg_percent_outcomes_by_states_by_tokens)
print(cooperative_epochs_by_states_by_tokens)
```

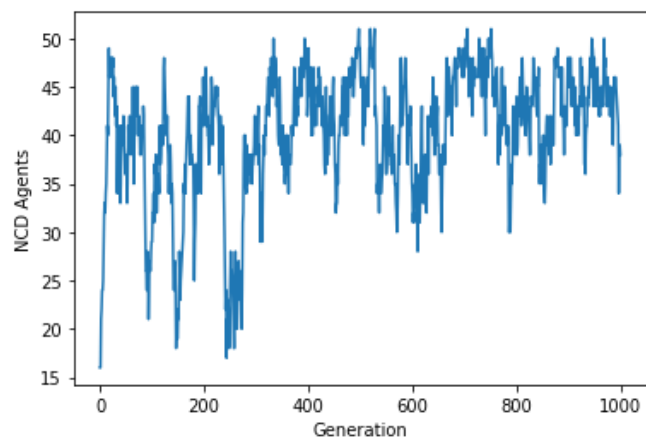
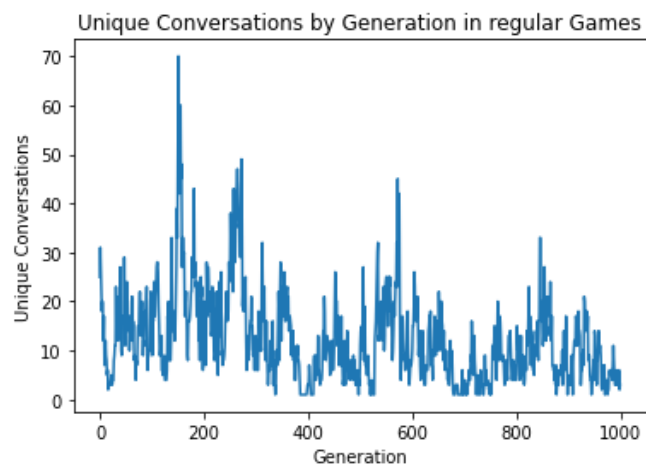
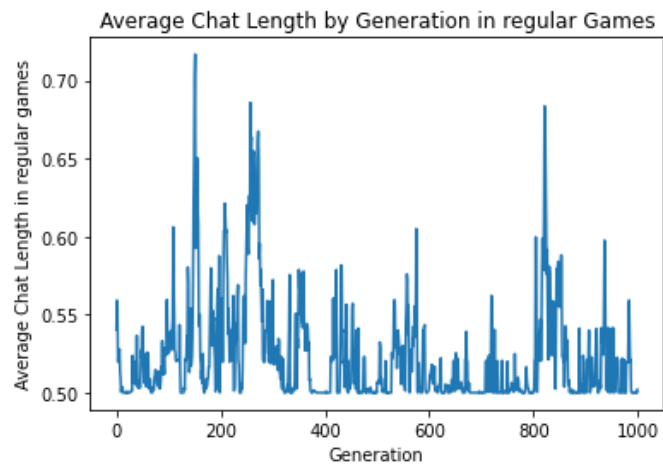
```
[[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014,
[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6]]]]
```

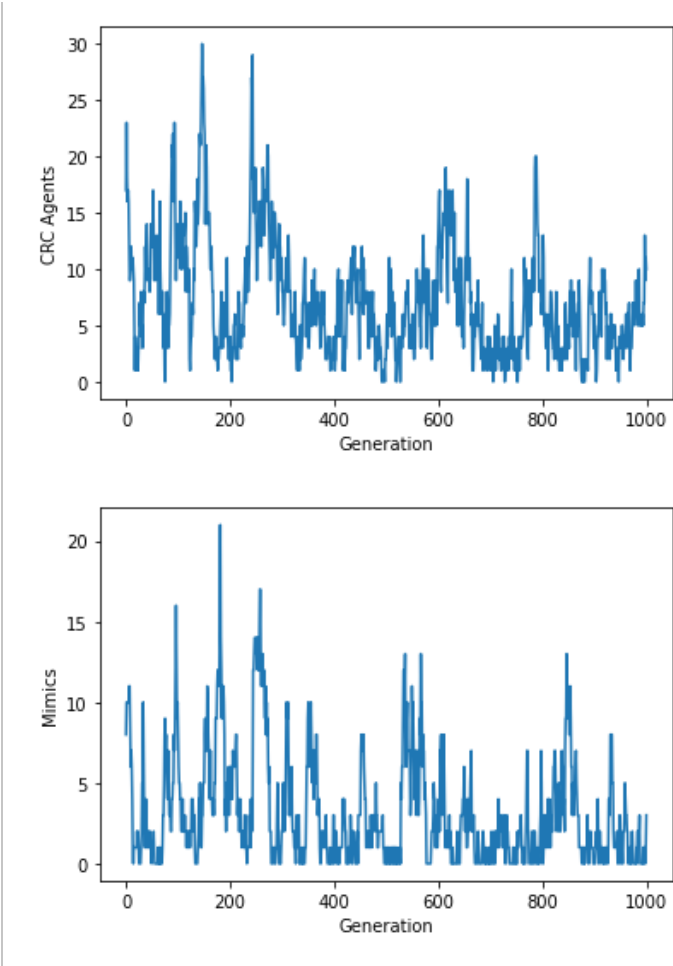
Sheet

```
###  
  
from Model import Model  
from GameHistoryAnalysis import GameHistoryAnalysis  
  
###  
  
number_of_agents = 50  
generations = 1000  
game = 'PD'  
learning_mechanism = 'GA'  
agent_computation_capacity = 4  
tokens = 3  
timeout = 100  
  
model = Model(number_of_agents, agent_computation_capacity, tokens, generations, learning_mechanism,  
model.run_model()  
  
###  
  
gha = GameHistoryAnalysis(model.game_history, number_of_agents)  
gha.plot_average_payoff()  
gha.plot_average_regular_game_payoff()  
  
###  
  
gha.plot_total_communication()  
gha.plot_total_regular_communication()  
  
###  
  
gha.plot_cooperation_percentage()  
  
###  
  
gha.plot_average_chat_length()  
gha.plot_average_regular_chat_length()  
  
###  
  
gha.plot_number_of_unique_conversations()  
  
###  
  
gha.plot_NCD_agents()  
  
###  
  
gha.plot_CRC_agents()  
  
###  
  
gha.plot_CD_agents()
```









Sheet 4

```

from Model import Model
from GameHistoryAnalysis import GameHistoryAnalysis
import matplotlib.pyplot as plt

number_of_agents = 50
generations = 1000
game = 'PD'
learning_mechanism = 'GA'
tokens = 4
timeout = 100
iterations = 5
states = 13

cooperative_epochs_by_states_by_tokens = [[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 5.2]]
avg_percent_outcomes_by_states_by_tokens = [[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014, 40.546580214796094], [9.528760023408973, 49.92465976179493, 40.546580214796094], [9.528760023408973, 49.92465976179493, 40.546580214796094]]]

for s in range(9, states):
    print('Starting Computation for computational capacity of ' + str(s))
    cooperative_epochs_by_tokens = []
    avg_percent_outcomes_by_tokens = []
    for t in range(tokens):
        print('Starting Computation for communication tokens: ' + str(t))
        total_coop_epochs = 0
        avg_percent_outcomes = [0, 0, 0]
        for i in range(iterations):
            print('Starting Iteration ' + str(i))
            model = Model(number_of_agents, s+1, t+1, generations, learning_mechanism, timeout, game)
            model.run_model()
            gha = GameHistoryAnalysis(model.game_history, number_of_agents)
            total_coop_epochs += gha.cooperative_epochs

            total = sum(gha.outcome_frequency)
            percent_outcomes = [(gha.outcome_frequency[m]/total) * 100 for m in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[j] + percent_outcomes[j] for j in range(3)]
            avg_percent_outcomes = [avg_percent_outcomes[k]/iterations for k in range(3)]
            avg_percent_outcomes_by_tokens.append(avg_percent_outcomes)
            cooperative_epochs_by_tokens.append(total_coop_epochs/iterations)
        avg_percent_outcomes_by_states_by_tokens.append(avg_percent_outcomes_by_tokens)
        cooperative_epochs_by_states_by_tokens.append(cooperative_epochs_by_tokens)
    print(avg_percent_outcomes_by_states_by_tokens)
    print(cooperative_epochs_by_states_by_tokens)

```

```

Starting Computation for computational capacity of 9
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 5.2]]
[[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014, 40.546580214796094], [9.528760023408973, 49.92465976179493, 40.546580214796094], [9.528760023408973, 49.92465976179493, 40.546580214796094]]]
Starting Computation for computational capacity of 10

```

```

Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6],
[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014,
Starting Computation for computational capacity of 11
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6],
[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014,
Starting Computation for computational capacity of 12
Starting Computation for communication tokens: 0
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 1
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
Starting Computation for communication tokens: 2
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3

```

```

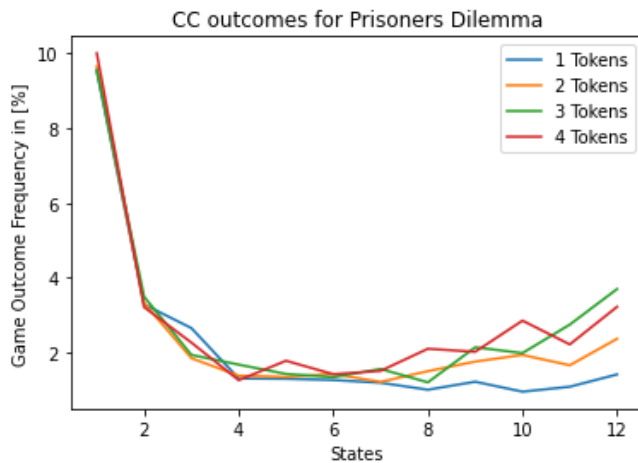
Starting Iteration 4
Starting Computation for communication tokens: 3
Starting Iteration 0
Starting Iteration 1
Starting Iteration 2
Starting Iteration 3
Starting Iteration 4
[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6],
[[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014,

```

```

import matplotlib.pyplot as plt
avg_percent_outcomes_by_states_by_tokens = [[9.528760023408973, 49.92465976179493, 40.546580214796094], [9.656737569023525, 49.593067041306014,
cooperative_epochs_by_states_by_tokens = [[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6]]
states = 12
tokens = 4
for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][0] for p in range(states)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('CC outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

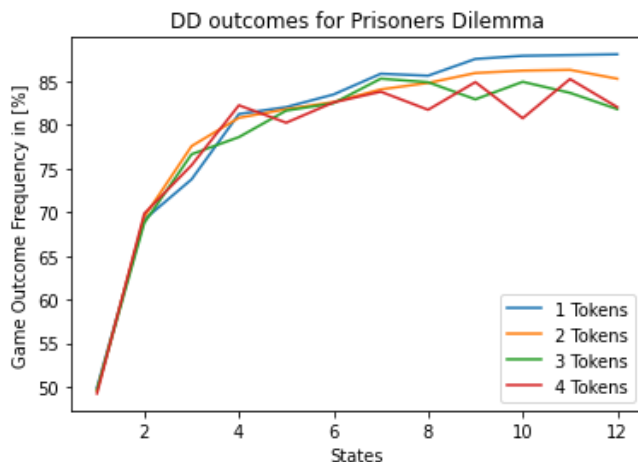
```



```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][1] for p in range(states)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DD outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

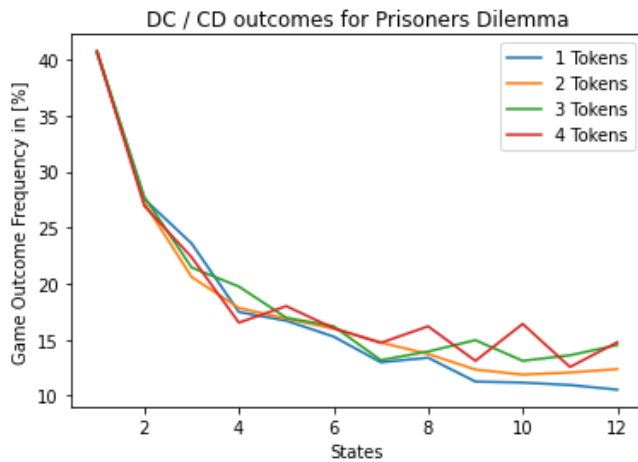
```



```

for t in range(tokens):
    plt.plot([s+1 for s in range(states)], [avg_percent_outcomes_by_states_by_tokens[p][t][2] for p in range(players)])
plt.ylabel('Game Outcome Frequency in [%]')
plt.xlabel('States')
plt.title('DC / CD outcomes for Prisoners Dilemma')
plt.legend()
plt.show()

```



```

print(cooperative_epochs_by_states_by_tokens)
for t in range(tokens):
    print(t)
    plt.plot([s+1 for s in range(states)], [cooperative_epochs_by_states_by_tokens[s][t] for s in range(players)])
plt.ylabel('Cooperative Epochs / 1000 Generations')
plt.xlabel('States')
plt.title('Total Cooperative Epochs / 1000 Generations for Stag Hunt')
plt.legend()
plt.show()

```

```

[[200.0, 198.8, 205.2, 220.4], [29.8, 35.2, 43.6, 32.8], [30.6, 13.0, 14.6, 20.4], [7.6, 9.0, 10.8, 7.6]]
0
1
2
3

```

