

```
In [ ]: #Below is a blueprint for how monte carlo simulations can be used for Money Line Betting
```

```
In [3]: import random

def simulate_game(team_a, team_b):
    # Determine the point spreads for the teams
    spread_a = team_a['point_spread']
    spread_b = team_b['point_spread']

    # Generate random scores for the teams
    score_a = random.randint(80, 120)
    score_b = random.randint(80, 120)

    # Adjust the scores based on the point spreads
    adjusted_score_a = score_a + spread_a
    adjusted_score_b = score_b + spread_b

    # Determine the winner of the game
    if adjusted_score_a > adjusted_score_b:
        winner = 'A'
    else:
        winner = 'B'

    # Return the simulated game results
    return {'team_a': team_a['name'], 'score_a': score_a, 'adjusted_score_a': adjusted_s
            'team_b': team_b['name'], 'score_b': score_b, 'adjusted_score_b': adjusted_s
            'winner': winner}

# Define the teams
team_a = {'name': 'Team A', 'point_spread': 3.5}
team_b = {'name': 'Team B', 'point_spread': -3.5}

# Simulate the game 1000 times
results = []
for i in range(1000):
    result = simulate_game(team_a, team_b)
    results.append(result)

# Calculate the win percentage for each team
win_count_a = sum([1 for result in results if result['winner'] == 'A'])
win_percentage_a = win_count_a / len(results)

win_count_b = sum([1 for result in results if result['winner'] == 'B'])
win_percentage_b = win_count_b / len(results)

# Print the win percentages
print(f'{team_a["name"]} win percentage: {win_percentage_a:.1%}')
print(f'{team_b["name"]} win percentage: {win_percentage_b:.1%}')
```

Team A win percentage: 66.6%

Team B win percentage: 33.4%