

Solução Teste Physa - Gabriel Vaz Cançado Ferreira

Modelagem do jogo:

i) Game class:

Esta é a classe principal que modela um jogo, ela armazena parâmetros do jogo como posição dos jogadores, contagem de cobras pisadas por cada jogador, rodada em que o jogo está, número de dados jogados por cada jogador. Além disso, temos o parâmetro `on`, usado na questão 4, e o parâmetro `immunity2` usado na questão 5. Além disso, a classe armazena as posições em que existem base de escadas e cabeças de cobra e para qual posição o jogador é deslocado caso caia em uma casa de escada ou cobra.

O método `throw_dice` usa a função pseudo aleatória `randint` da biblioteca `random` para simular um lançamento de um dado. O método `play_round` é responsável por cada rodada. Ela permite que cada jogador jogue o dado, mova, chama o método que checa se é uma casa de escada ou cobra e checa se o jogador venceu. Caso um jogador vença, ela retorna o jogador vencedor, caso contrário apenas retorna 0.

O método `check_stairs_and_snakes` confere se o jogador está em uma casa de escada ou cobra e o desloca de acordo. Além disso, ela tem especificidades das soluções de algumas questões. O método `stair_probabilities` é usado na questão 3 para adicionar um fator de probabilidade na subida de escadas.

O construtor da classe `Game` recebe as posições iniciais dos jogadores e a opção de debug.

ii) runGame function:

Função simples que cria um loop de rounds até que algum dos jogadores vença. É também usada para contar o número de rounds por jogo. Retorna o vencedor (1 ou 2) e o objeto da classe `game` ao final do jogo.

Solução das questões

- 1) Implementação da função `winningProbability` que itera sobre 10000 jogos para descobrir a probabilidade de vitória de cada jogador. Ao executar o arquivo, há a execução de 1000 simulações para determinar um range de probabilidades estimado para o qual há significância os valores de probabilidade de vitória do jogador 1. O valor mais provável é de 52,25% de chance de vitória do jogador 1.
- 2) Implementação da função `count_nakes` que itera sobre 10000 jogos para contar quantas cobras cada jogador pisou. Ao executar o arquivo, são realizadas 1000 simulações para encontrar um range de valores de confiança para o número médio de cobras pisadas por cada jogador. Temos que para o jogador 1 o valor mais provável está entre 1.58 e 1.60 e para o jogador 2 esse valor é de 1.50. No final, os resultados

são salvos em um arquivo csv que são lidos no notebook 2.ipynb por comodidade devido ao tempo de execução dessa simulação ser maior.

- 3) Implementação da função `count_dice_throws` que executa uma simulação de 10000 jogos e conta quantos lançamentos de dados são feitos por cada jogador. Ao executar o arquivo é feita uma simulação de 1000 experimentos para determinação de um range de confiança para o número médio de dados lançados. Note que para essa questão o parâmetro `on` está igual a `True` na classe `Game`, para que seja adicionado fator de probabilidade de 50% dos jogadores subirem ou não uma escada, o que é feito pela função `stair_probabilities`. Para ambos os jogadores o número médio de lançamento de dados é de 11 lançamentos para completar o jogo. Os resultados são salvos em arquivo csv e podem ser vistos no arquivo 3.ipynb.
- 4) Para resolver essa questão, repetimos a função usada na questão 1, porém agora passamos como parâmetro inicial a posição inicial do jogador 2 e retorna o valor da posição inicial em que a probabilidade de vitória dos jogadores difere de menos de 1% . Ao executar o arquivo, realizamos 100 simulações em que em cada uma todos os valores de posição inicial são avaliados e registramos as casas em que a probabilidade de vitória dos indivíduos esteve muito próxima. Por fim, usamos a função `count_numbers` para contar quais valores mais apareceram. A casa 7, pode ser definida, como a casa para o jogador 2 inicial que mais aproxima a probabilidade de vitória dos jogadores.
- 5) Repetimos a questão 1 mas tornando o parâmetro `immunity2` `True` na classe `Game`. Assim, chegamos que caso o jogador 2 tenha imunidade a primeira cobra, a probabilidade de vitória do jogador 1 cai para 27.5%.