

## Trabalho 07 - Destruindo torres

Data de entrega: 20/12/2021

Importante:

- **Não** olhe códigos de outros ou da internet, exceto o que é fornecido. Também não mostre ou publique o seu.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Você deverá submeter, até a data de entrega, o seu código na plataforma run.codes. Seu código deverá rodar em até 1 segundo no run.codes para cada instância.

Tower Attack 3000 é um jogo de “tower attack” em que você vai atacar torres que o inimigo está conjurando para te atacar. O jogo se dá em turnos, e o número de turnos é definido no início do jogo. No campo de batalha o inimigo pode conjurar até 5000 torres. Cada turno é composto das seguintes ações:

1. Se couber no campo de batalha, o inimigo conjura uma nova torre. A quantidade de pontos de vida (hit points - hp) da torre é sorteado aleatoriamente entre 1 e 20. Cada torre gerada tem um id sequencial, ou seja, a primeira torre tem id 0, a segunda tem id 1 e assim por diante.
2. O Jogador pode atacar uma torre, tirando 10 pontos de vida dela. De a torre ficar com 0 ou menos de vida ela cai e sai do jogo.
3. As torres que permanecem vivas no final do turno atacam o jogador, cada torre causa 1 de dano.

O objetivo do jogador é minimizar o seu dano. Para isso, uma estratégia interessante é atacar sempre a torre com a menor quantidade de vida, se houverem duas torres com a mesma quantidade de vida, o jogador decidiu que vai atacar aquela a mais tempo no jogo, ou seja, aquela com id menor. Seu objetivo neste trabalho é implementar essa estratégia.

A partir da próxima página veremos uma simulação do jogo com 3 turnos.

TURNO 1: No primeiro turno o inimigo conjura uma torre (a torre com id 0) com 13 pontos de vida. Logo após o jogador ataca a torre com menor quantidade de vida (como só tem uma torre essa decisão é bem simples), retirando 10 pontos de vida dela. A torre 0 sobrevive com 3 pontos de vida, e portanto ao final do turno ela pode atacar o jogador, lhe causando 1 de dano.



TURNO 2: No segundo turno o inimigo conjura uma nova torre (a torre 1) com apenas 2 pontos de vida. A próxima ação é do jogador e ele ataca a torre com menos pontos de vida, nesse caso a torre 1, retirando 10 pontos de vida dela. Como o total de pontos de vida da torre 1 ficaria negativo (-8) ela cai e sai do jogo. No final do turno a torre 0, que ainda está em pé, ataca o jogador lhe causando mais 1 de dano.



TURNO 3: O inimigo conjura uma torre com 16 pontos de vida. O jogador ataca a torre 0, destruindo ela. A torre 2 ataca o jogador causando 1 de dano.



O jogo acaba e o resultado é que o jogador tomou 3 de dano.

A entrada é apenas o número de turnos e o jogo é

4

A saída está pré-definida na função main que será disponibilizada. Cada ação é impressa na tela: A conjuração de uma nova torre, o ataque do jogador, a queda de uma torre e o dano causado ao jogador.

Nova torre 0: 2

Atk 0: 2 -> -8

t0 caiu

0 dano

Nova torre 1: 4

Atk 1: 4 -> -6

t1 caiu

0 dano

Nova torre 2: 15

Atk 2: 15 -> 5

1 dano

Nova torre 3: 7

Atk 2: 5 -> -5

t2 caiu

1 dano

FINAL:

Jogador tomou 2 dano

- Você receberá um código com a função main que você poderá usar, você pode alterá-la se quiser, desde que isso não mude a saída que deve ser gerada (o que inclui a ordem com que as torres são geradas). Você precisará implementar a fila de prioridades que vai guardar as torres e indicar qual será a próxima a ser atacada.