

Zasady Turnieju HackArena 2.0

Waszym zadaniem jest napisanie programów (botów, agentów) do stworzonej przez nas gry. Rywalizacja będzie się opierać na turniejowym drzewku zaczynając od ćwierćfinałów. Na początku podzielimy wszystkie 16 drużyn na 4 grupy, więc każda grupa będzie miała po 4 drużyny. Te 4 grupy tworzą 4 mecze ćwierćfinałowe. W jednym meczu biorą udział 4 drużyny, czyli 4 boty. Mecz polega na rozegraniu 5 rund. Z każdego meczu 2 najlepsze boty awansują do następnego etapu. Przy czym w finale oczywiście zwycięzca jest tylko jeden, a kolejne drużyny otrzymają odpowiednio drugie, trzecie i czwarte miejsce. A więc spytacie jak wyłaniamy zwycięzców w meczach?

System oceniania w meczu

Każdy z 4 graczy (A, B, C, D) grających mecz z wieloma rundami otrzymuje punkty na podstawie ich ostatecznych wyników oraz ilości zabójstw jakie zrobił dany gracz w danej rundzie.

Poprzez "wynik w rundzie" rozumiemy ilość punktów zdobytych w grze za zadawanie obrażeń oraz przejmowanie stref. A poprzez "punkty" rozumiemy zbierane przez boty punkty zbierane w całym meczu za zajęte miejsca w rundach.

- 1. miejsce (783 wyniku i 3 zabicia) otrzymuje 3 punkty
- 2. miejsce (457 wyniku i 2 zabicia) otrzymuje 2 punkty
- 3. miejsce (457 wyniku i 1 zabicia) otrzymuje 1 punkt
- 4. miejsce (122 wyniku i 3 zabicia) otrzymuje 0 punktów

Przykład:

Założmy dla uproszczenia, że ci gracze grają 3 rundy (w rzeczywistości będzie ich 5).

Poniżej przedstawiono, jak przyznawane są punkty po każdej rundzie.

Wyniki Rundy 1:

- 1. miejsce: Gracz A (3 punkty)
- 2. miejsce: Gracz C (2 punkty)
- 3. miejsce: Gracz D (1 punkt)
- 4. miejsce: Gracz B (0 punktów)

Wyniki Rundy 2:

- 1. miejsce: Gracz B (3 punkty)
- 2. miejsce: Gracz A (2 punkty)
- 3. miejsce: Gracz C (1 punkt)
- 4. miejsce: Gracz D (0 punktów)

Wyniki Rundy 3:

- 1. miejsce: Gracz C (3 punkty)
- 2. miejsce: Gracz D (2 punkty)
- 3. miejsce: Gracz B (1 punkt)

- 4. miejsce: Gracz A (0 punktów)

Ostateczna Tabela Wyników:

- Gracz A: 3 (Runda 1) + 2 (Runda 2) + 0 (Runda 3) = 5 punktów
- Gracz B: 0 (Runda 1) + 3 (Runda 2) + 1 (Runda 3) = 4 punkty
- Gracz C: 2 (Runda 1) + 1 (Runda 2) + 3 (Runda 3) = 6 punktów
- Gracz D: 1 (Runda 1) + 0 (Runda 2) + 2 (Runda 3) = 3 punkty

Po rozegraniu wszystkich rund, Gracz C ma najwięcej punktów i zostaje ogólnym zwycięzcą meczu. Drugie miejsce zajmuje Gracz A. Te 2 boty awansują do kolejnego etapu.

Przypadki szczególne

1. Remisy

W przypadku, gdy dwóch lub więcej graczy będzie miało taki sam wynik w rundzie (mało prawdopodobne, ale jednak), gracze z tym samym wynikiem otrzymują punkty za miejsce wyższe. Na przykład, jeśli Gracz A i Gracz B zajmą 1. miejsce w rundzie, obaj dostaną po 3 punkty, a kolejny gracz zostanie sklasyfikowany na 3. miejscu i otrzyma 1 punkt (ponieważ 2. miejsce jest pomijane). Zapewnia to sprawiedliwy podział punktów w przypadku remisów, a gracze są jednakowo nagradzani za podobne wyniki. W przypadku remisu na 2. miejscu, obaj gracze otrzymują po 2 punkty, a 4. gracz automatycznie otrzymuje 0 punktów.

2. Gracz, który nic nie zrobił

Jeśli jakikolwiek gracz nie zrobi nic w rundzie — to znaczy, że ma wynik 0 z rundy, innymi słowy zgromadzi 0 punktów w grze — taki gracz automatycznie otrzymuje 0 punktów, niezależnie od zajmowanej pozycji. Taka sytuacja mogłaby mieć miejsce tylko gdy jest kilku graczy, którzy by mieli wynik 0 z rundy.

3. Trzech graczy ex aequo w meczu po wszystkich rundach (tyle samo punktów i zabić)

W takim przypadku, rozgrywamy dodatkowe rundy do momentu, gdy dwóch graczy będzie ewidentnymi zwycięzcami. A w przypadku finału do momentu gdy wszystkie 4 miejsca będą jednoznaczne.

Jak będziemy testować wasze boty?

1. Poprosimy was o spakowanie zmodyfikowanego repozytorium do pliku .zip. Oczywiście prosimy o usunięcie wszystkich plików i folderów tymczasowych zawierających zbudowane .exe czy .jar, czy node_modules, aby plik zip nie był zbyt duży.
2. Po zebraniu wszystkich paczek zip od was, przetrzucimy je na nasz serwer.
3. Napisaliśmy parę skryptów w języku Python wraz z biblioteką Docker SDK for Python, które w pierwszej kolejności pozwolą nam na zbudowanie obrazów dockera dla każdego z

waszych botów. Dlatego ważne jest, abyście nie modyfikowali żadnych plików poza wskazanymi plikami i folderami w README każdego z repozytoriów.

4. Następnie kolejny skrypt uruchomi odpowiednio kontener z serwerem, jak i klientami (wasze boty), aby rozegrać mecze. I tak podobnie dla całej drabinki turniejowej.