

<⚡> HackArena 2.0

Zasady Hackathonu

Witajcie na HackArena 2.0!

Waszym zadaniem jest **napisanie programów (botów)** do stworzonej przez nas gry **MonoTanks**. Rywalizacja będzie się opierać na turniejowym drzewku zaczynając od ćwierćfinałów.

Na początku zostaliście podzieleni na **4 grupy po 4 drużyny**. Każda grupa rozegra w trakcie turnieju **mecz ćwierćfinałowy**. W jednym meczu biorą udział **4 drużyny, czyli 4 boty**.

Mecz polega na rozegraniu **10 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. Po więcej informacji jak wygląda ocenianie botów w meczu proszę spojrzeć na rozdział System oceniania w meczu poniżej.

Aby **zwyciężyć w HackArena 2.0** Wasz bot, będzie musiał nie tylko bezbłędnie **implementować podstawowe mechaniki**, ale także będzie musiał wykazać się **"myśleniem strategicznym"**. Pomyślcie o tym w trakcie implementacji Waszego bota! Pamiętajcie też, że w trakcie meczu, może wydarzyć się wiele **sytuacji brzegowych** (tzw. edge case) - im więcej z nich wyłapiecie w trakcie tworzenia bota - tym mniejsza szansa, że coś go zaskoczy w trakcie trwania turnieju!

Szczegółowe informacje na temat gry, systemu oceniania oraz tego jak **zacząć pisać bota** znajdziecie poniżej w tym pliku. Zapoznajcie się z nimi dokładnie!

Miłej lektury ;)

Instrukcja gry

Instrukcja zawiera jedynie **podstawowe informacje** na temat zasad gry i **nie opisuje wszystkich możliwych sytuacji**, które mogą wystąpić podczas rozgrywki. Przedstawione tu mechaniki mają na celu wprowadzenie w fundamenty gry, jednak pełne zrozumienie synergii i nietypowych przypadków **wymaga samodzielnego eksperymentowania**. Zachęcamy do dalszego testowania i poznawania złożoności gry, gdyż może mieć istotny wpływ na Wasze osiągnięcia i wynik rozgrywki.

Wstęp

MonoTanks to wieloosobowa gra od 2 do 4 graczy, gdzie każdy gra przeciwko każdemu. Polega ona na sterowaniu czołgiem, który ma za zadanie zdobywać punkty zadając obrażenia oraz przejmując strefy. Do realizacji tego celu może wykorzystać kilka rodzajów broni.

Mechaniki

Czołgi

Czołg to jednostka w grze, którą steruje bot. Ma on 100 punktów życia. Ma on podstawową broń, którą jest pocisk podstawowy, oraz może mieć w ekwipunku maksymalnie jedną broń specjalną.

Gra odbywa się w prostej pętli. Serwer wysyła aktualny stan gry, a bot może na niego zareagować i wykonać jedną z następujących akcji:

1. Obrót wieży o 90 stopni w lewo, w prawo lub brak obrotu i jednocześnie obrót czołgu o 90 stopni w lewo, w prawo lub brak obrotu. (Obrót wieży nie wpływa na obrót czołgu ani na odwrót)
2. Poruszenie się o jedną kratkę w przód lub w tył w kierunku wskazywanym przez podwozie czołgu.
3. Strzał z broni podstawowej lub użycie broni specjalnej.
4. Czekanie (nie robienie nic).

<⚡> HackArena 2.0

Mgła wojny

Bot widzi tylko pola, które znajdują się w stożku 144 stopni przed czołgiem. Nie ma ograniczenia odległości. Dodatkowo bot widzi pola w linii prostej przed wieżą czołgu. Wszystkie inne pola zakrywane są przez mgłę wojny. Mgła wojny nie dotyczy za to ścian oraz stref. Oczywiście w strefie, która jest w mgle wojny, nadal nie widać czołgów innych botów.

Bronie

Pocisk podstawowy

Leci on z prędkością 2 kratek na tick (jeden tick to jedna aktualizacja stanu gry) i zadaje 20 obrażeń przy trafieniu. Amunicja ta regeneruje się pasywnie z prędkością 1 pocisk na 10 ticków. Czołg może przenosić maksymalnie 3 sztuki tego pocisku. Pociski te przy zderzeniu ze sobą anihilują się wzajemnie i po prostu znikają.

Bronie specjalne

Czołg nie posiada domyślnie przy sobie żadnej broni specjalnej. Aby ją zdobyć należy znaleźć ją na mapie i najechać na nią. Wtedy trafi ona do ekwipunku czołgu. Czołg może przenosić maksymalnie jedną broń specjalną, a w przypadku gdy już ją posiada i najedzie na kolejną broń specjalną, to nowej broni po prostu nie można będzie wziąć dopóki nie wykorzysta tej którą posiada.

Dubeltówka

Jest to broń specjalna, która wystrzeliwuje dwa równoległe pociski jednocześnie. Zadają one łącznie 2 razy więcej obrażeń niż pocisk podstawowy, czyli 40 obrażeń przy trafieniu.

Mina

Gdy bot użyje miny, jest ona stawiana na ziemi jedną kratkę za czołgiem. Mina zadaje 50 obrażeń gdy ktokolwiek na nią wjedzie.

<⚡> HackArena 2.0

Laser

Laser jest najsilniejszą bronią w grze. Jego użycie oznacza natychmiastowe rażenie wszystkiego co znajduje się na jego drodze, dopóki nie napotka ściany. Jego użycie trwa 10 ticków, to oznacza, że strzelec jest unieruchomiony na 10 ticków i nie może podjąć żadnej innej akcji. Zadaje on 80 obrażeń przeciwnikowi przy trafieniu w każdym ticku kiedy przeciwnik stoi w wiązce lasera.

Radar

Użycie radaru powoduje, że w następnym ticku widzimy jednorazowo całą mapę pomimo mgły wojny, a w szczególności czołgi wszystkich botów oraz bronie specjalne, które znajdują się na mapie.

Punkty za obrażenia

Boty otrzymują punkty za zadawane obrażenia innym botom. Otrzymują one różną ilość punktów w zależności od użytej broni. Proszę spojrzeć na tabelę broni poniżej.

Dodatkowo, w sytuacji gdy zadamy przeciwnikowi większe obrażenia niż posiada on punktów zdrowia, to dostajemy odpowiednio tyle punktów jakbyśmy mu zadali tyle obrażeń ile miał punktów zdrowia. Czyli przykładowo jeśli przeciwnik posiada 20 punktów życia i został zniszczony dubeltówką przez Waszego bota, to otrzymacie 10 punktów.

Tabela broni

Broń	Mnożnik punktów	Obrażenia (Zdobyte punkty)	Prędkość (kratka/tick)	Czas strzelania (tick)
Pocisk podstawowy	0.5x	20 (10)	2	1
Dubeltówka	0.5x	40 (20)	2	1
Laser	1x	80 (80)	inf	10
Mina	1x	50 (50)	N/A	1

<⚡> HackArena 2.0

Strefy

Na mapie są zawsze 2 strefy. Każda strefa może być w jednym z 5 stanów:

1. Neutral - żaden bot nie posiada tej strefy.
2. BeingCaptured - strefa jest właśnie przejmowana przez jakiegoś bota. To znaczy w strefie znajduje się tylko jeden bot i rośnie jego posiadanie strefy lub bot przejmujący wyszedł z niej i jego posiadanie się zmniejsza. Przejęcie strefy trwa 50 ticków.
3. Captured - strefa jest przejęta przez kogoś. Tylko w tym stanie bot dostaje punkty za posiadanie strefy. 1 punkt co drugi tick.
4. BeingContested - w strefie znajduje się więcej niż jeden bot. Strefa nie jest przez nikogo przejmowana, dopóki jest więcej niż jeden bot w niej. Jeśli ktokolwiek przejął wcześniej strefę, to w tym stanie bot nie dostaje punktów za posiadanie strefy.
5. BeingRetaken - strefa została przejęta przez innego bota wcześniej. Teraz jest przejmowana przez innego bota. Przejęcie strefy trwa 50 ticków. W tym stanie poprzedni bot posiadający strefę oczywiście nie dostaje punktów za posiadanie strefy.

Odrodzenie się

Czołg po zniszczeniu znika z mapy i zostanie na nią zrzucony po 50 tickach.

Leczenie

Czołg leczy się o 40 punktów życia po zniszczeniu przeciwnika oraz do maksymalnie 80 punktów życia (1 punkt co 4 ticki) podczas przebywania w przejętej przez siebie strefie.

Informacje Techniczne

Aby mieć pewność, że Wasze boty **będą działać poprawnie** na naszym serwerze, możecie w każdej chwili nas poprosić, abyśmy **przetestowali** Waszą **obecną wersję** na systemie sprawdzającym, czy na pewno wszystko się **kompiluje, buduje, uruchamia** i czy wyniki są w porządku. Jeśli na koniec hackathonu Wasz bot **nie będzie działał poprawnie** (nie zbuduje się obraz dockera, nie uruchomi się, będzie się crashował), to **Wasz bot zostanie zdyskwalifikowany** i turniej rozegra się bez niego.

Jak zacząć pisać bota?

Dla każdego języka programowania będzie to wyglądało nieco inaczej. Ogólnie rzecz biorąc, cały proces wygląda następująco:

1. Najpierw musicie uruchomić lokalnie serwer gry. W tym celu wejdźcie na repozytorium <https://github.com/INIT-SGGW/HackArena2.0-MonoTanks>, gdzie w pliku README znajdziecie instrukcje, jak uruchomić serwer na kilka sposobów.
2. Następnie wejdźcie na repozytorium klienta w wybranym przez was języku programowania i przeczytajcie jego README, aby dowiedzieć się, jak uruchomić klienta oraz jak napisać bota.

<⚡> HackArena 2.0

Dostępne języki programowania

1. Python
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-Python>
2. Java
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-Java>
3. JavaScript
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-JS>
4. TypeScript
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-TS>
5. C#
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-CSharp>
6. C++
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-Cxx>
7. Rust
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-Rust>
8. GO
<https://github.com/INIT-SGGW/HackArena2.0-MonoTanks-Go>

Architektura systemu

Stosujemy klasyczną architekturę klient-serwer, dzięki czemu możecie pisać bota w dowolnym języku programowania. Komunikacja sieciowa odbywa się przy użyciu WebSocketów.

- Serwer - Zajmuje się on zarówno sieciowym aspektem połączenia ze sobą klientów (graczy i botów), jak i logiką gry.
- Klient - Może to być klient GUI za pomocą którego gracz może obserwować rozgrywkę lub samemu grać, albo też klient-bot, którego wy piszecie.

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, lub innych niepotrzebnych plików i folderów jak .git, aby plik zip nie był zbyt duży.
2. Po zebraniu wszystkich paczek zip od was, przetrzucimy je na nasz serwer.
3. Napisaliśmy skrypt w języku Python wraz z biblioteką Docker SDK for Python, który w pierwszej kolejności pozwoli 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 skrypt dla każdej rundy każdego meczu uruchomi odpowiednio kontener z serwerem, jak i klientami (Wasze boty), aby rozegrać mecze, zgodnie z zasadami drzewka turniejowego, zapisując wszelkie pliki wynikowe, powtórki i logi.

W ten sposób przeprowadzimy symulację turnieju, rozgrywając dla każdego meczu 10 rund. Pozwoli nam to sprawdzić, który z Waszych botów jest faktycznie najlepszy.

FAQ

1. Czy mogę używać zewnętrznych bibliotek/narzędzi do pisania botów?

→ W 99% przypadków tak. Jeśli biblioteka nie powoduje efektów ubocznych wpływających na system operacyjny, sieć, kartę graficzną lub inne tego typu elementy, to wszystko powinno być w porządku. Wszelkie proste biblioteki do analizy danych czy łatwiejszego przetwarzania tablic są jak najbardziej dozwolone. Oczywiście nie wolno usuwać ani zmieniać wersji bibliotek, które są obecnie wykorzystywane przez repozytoria, ponieważ może to wpłynąć na kompatybilność z naszym serwerem i systemem sprawdzania botów.

2. Czy bot może wykonywać zapytania do zewnętrznych serwerów w trakcie gry?

→ Nie, nie może. W środowisku dockera na serwerze kontenery nie będą miały dostępu do internetu.

3. Czy mogę w repozytorium umieścić pliki na przykład wag sieci neuronowych?

→ Tak, powinno być wszystko w porządku. W każdym z API wrapperów w README jest opisane, jak załączać statyczne pliki.

<⚡> HackArena 2.0

4. Chciałbym napisać bota z użyciem tensorflow i użyć karty graficznej. Czy mój bot podczas sprawdzania będzie mógł korzystać z karty graficznej?

→ Nie, nie będzie mógł. Boty uruchamiamy jako kontenery dockera na serwerze, który nie ma żadnej karty graficznej.

5. Ile pamięci RAM będzie dostępne mojemu botowi podczas gry?

→ Dostępny będzie 1GB RAMu na jednego bota.

6. Ile wątków procesora będzie dostępne mojemu botowi podczas gry?

→ Dostępny będzie 1 wątek procesora na jednego bota.

7. Jak duże moje repozytorium może być i jak dużo przestrzeni dyskowej będzie dostępne dla mojego bota?

→ Im mniejsze tym lepsze. Aczkolwiek jeśli ktoś potrzebuje dodać pliki statyczne to maksymalnie repozytorium po rozpakowaniu może mieć 1GB.

8. Czy mogę uruchamiać asynchroniczne funkcje w moim rozwiązaniu?

→ Zależy od implementacji konkretnego API wrappera. Niektóre na to pozwalają, niektóre nie.

System oceniania

Turniej finałowy HackArena 2.0 zostanie przesymulowany automatycznie i botom zostaną przydzielone punkty zgodnie z poniższymi zasadami.

Aby zagwarantować jak najbardziej rzetelne wyniki, każdy mecz zostanie rozegrany w 10 rundach, a sumaryczne wyniki zadecydują o kolejności drużyn w meczu. Tak więc mówiąc **"mecz"** mamy na myśli dziesięciokrotne przesymulowane rozgrywki. Taka pojedyncza symulacja to właśnie **"runda"** w meczu.

Każdy z 4 botów (A, B, C, D) grających mecz z wieloma rundami otrzymuje punkty na podstawie ostatecznych wyników oraz ilości zabójstw jakie zrobił dany bot 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. Najpierw sortujemy boty według wyniku, a w drugiej kolejności według zabić. Przykładowo:

- 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

Przykładowy mecz:

Założmy dla uproszczenia, że boty grają 3 rundy (w rzeczywistości będzie ich 10). Poniżej przedstawiono, jak przyznawane są punkty po każdej rundzie.

Wyniki Rundy 1:

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

<⚡> HackArena 2.0

Wyniki Rundy 2:

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

Wyniki Rundy 3:

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

Ostateczna Tabela Wyników:

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

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

Przypadki szczególne

1. Remisy

W przypadku, gdy dwa lub więcej botów będzie miało taki sam wynik w rundzie oraz tyle samo zabić (mało prawdopodobne, ale jednak), boty z tym samym miejscem otrzymują punkty za miejsce wyższe. Na przykład, jeśli Bot A i Bot B zajmą 1. miejsce w rundzie, obaj dostaną po 3 punkty, a kolejny bot zostanie sklasyfikowany na 3. miejscu i otrzymuje 1 punkt (ponieważ 2. miejsce jest pomijane).

W przypadku remisu na 2. miejscu, oba boty otrzymują po 2 punkty.

<⚡> HackArena 2.0

2. Bot, który nic nie zrobił

Jeśli jakikolwiek bot nie zrobi nic w rundzie – to znaczy, że ma wynik 0 z rundy, innymi słowy zgromadzi 0 punktów w grze – taki bot automatycznie otrzymuje 0 punktów, niezależnie od zajmowanej pozycji. Taka sytuacja mogłaby mieć miejsce tylko gdy jest kilka botów, które by miały wynik 0 z rundy.

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

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

4. W meczu uczestniczą mniej niż 4 boty

Taka sytuacja może nastąpić w przypadku gdy któraś drużyna zostanie zdyskwalifikowana. W tym przypadku jedyne co się zmienia to punktacja. W przypadku trzech botów pierwsze miejsce otrzymuje 2 punkty, drugie 1 punkt, a trzecie 0 punktów. W przypadku dwóch botów pierwsze miejsce otrzymuje 1 punkt, drugie 0 punktów.

W przypadku gdy w meczu jest tylko jeden bot (jego przeciwnicy nie dojechali :P), mecz nie jest rozegrany. I bot automatycznie awansuje do następnego etapu.



Powodzenia!

W przypadku jakichkolwiek problemów zgłaszajcie się do organizatorów HackArena 2.0 - z pewnością pomożemy!

Życzymy dobrej zabawy i zwycięstwa w drugiej edycji hackathonu HackArena!

Zespół HackArena
Członkowie KN init