

Hack⚡Arena 2.5

Zasady Hackathonu

Witajcie na HackArena 2.5!

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

Na początku zostaliście podzieleni na **4 grupy po 2 drużyny**. Każda grupa rozegra w trakcie turnieju **mecz ćwierćfinałowy**. W jednym meczu biorą udział **2 drużyny, czyli 2 boty***.

Mecz polega na rozegraniu **9 rund (Best of 9)**. Z każdego meczu **najlepszy bot awansuje** do następnego etapu. Po więcej informacji jak wygląda ocenianie botów w meczu proszę spojrzeć na rozdział System oceniania w meczu poniżej.

Aby **zwygrać** w HackArena 2.5 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 ;)

Hack⚡Arena 2.5

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

StereoTanks to wieloosobowa gra w której 2 drużyny rywalizują ze sobą. Każda drużyna posiada 2 boty (czołgi) różnego typu: lekki (light) oraz ciężki (heavy). Polega ona na sterowaniu wspomnianymi czołgami, które mają za zadanie zdobywać punkty zadając obrażenia oraz przejmując strefę. Do realizacji tego celu może wykorzystać kilka rodzajów broni.

PS. Implementujecie jednego bota (jedno repozytorium), który w zależności od argumentu konsolowego będzie pełnił rolę czołgu lekkiego lub ciężkiego.

Mechaniki

Czołgi

Czołg to jednostka w grze, którą steruje bot. Ma on 100 punktów życia. Ma on kilka broni (pocisków): pocisk podstawowy, pocisk stunujący oraz pocisk leczący. W zależności od typu czołgu posiada także:

- podwójny pocisk oraz radar dla czołgu lekkiego,
- laser i minę dla czołgu ciężkiego.

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

Hack⚡Arena 2.5

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. Ruch w kierunku wskazanego pola. **Innymi słowy jest to *pathfinding*.**
5. Przejęcie udziałów w strefie.
6. Czekanie (nie robienie nic).

Mgła wojny

Pojedynczy czołg 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 przeciwników. Ponadto czołgi w jednej drużynie widzą swoje wzajemne pola widzenia.

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. 10

Bronie specjalne

Czołgi posiadają różne bronie w zależności od tego czy to czołg lekki czy ciężki. Oba typy posiadają pocisk stunujący oraz pocisk leczący. Czołg lekki posiada pocisk podwójny (Dubeltówka) oraz radar. Czołg ciężki posiada laser oraz minę. Czołgi po odrodzeniu mają od razu wszystkie umiejętności dostępne. Te bronie po użyciu odnawiają się w ekwipunku czołgu i po pewnej ilości ticków znów są dostępne do użycia.

Hack⚡Arena 2.5

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. Odnawia się po 100 tickach po użyciu.

Mina

Gdy bot użyje miny, jest ona stawiana na ziemi jedną kratkę za czołgiem. Mina zadaje 35 obrażeń gdy ktokolwiek na nią wjedzie. Odnawia się po 80 tickach po użyciu.

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.

Ponadto laser jest jedyną bronią, która może zniszczyć minę. Pozostałe pociski przelatują nad miną i nie niszczą jej. Odnawia się po 400 tickach po użyciu.

Radar

Użycie radaru powoduje, że w następnym ticku widzimy jednorazowo całą mapę pomimo mgły wojny, czyli w szczególności czołgi przeciwnika oraz miny, które znajdują się na mapie. Odnawia się po 200 tickach po użyciu.

Pocisk Stunujący

W zależności od czołgu strzelającego działa on trochę inaczej. Gdy czołg lekki go używa to czołg trafiony nie może używać żadnej broni w tym radaru przez 10 ticków. Gdy czołg ciężki go używa to trafiony czołg nie może się ruszać do przodu lub do tyłu przez 10 ticków, ale nadal może obracać czołgiem i wieżyczką. Odnawia się po 200 tickach po użyciu.

Hack⚡Arena 2.5

Pocisk Leczący

Pocisk leczący gdy trafi w czołg leczy go o 20 punktów życia. Oczywiście maksymalna liczba punktów życia to 100. Odnawia się po 50 tickach po użyciu.

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 (ticks)	Czas odnowienia (ticks)
Pocisk podstawowy	0.5x	20 (10)	2	1	10
Dubeltówka	0.5x	40 (20)	2	1	60
Laser	1x	80 (80)	inf	10	400
Mina	1x	35 (35)	N/A	1	100
Radar	N/A	N/A	N/A	1	200
Pocisk Stunujący	N/A	N/A	2	1	200
Pocisk leczący	N/A	N/A	2	1	50

Hack⚡Arena 2.5

Strefy

Na mapie jest zawsze jedna strefa. Strefa działa trochę jak firma w której można mieć udziały. W zależności od tego jak dużo udziałów ma drużyna to tak dużo punktów dostaje z każdym tickiem. Początkowo udziały strefy są neutralne i nie należą do nikogo, a więc nikt nie dostaje punktów. Aby przejąć część udziałów strefie, czołg musi być w strefie i wykonać akcję przejęcia strefy (CaptureZone). Ta akcja bota przejmie tylko mały ułamek udziałów. Konkretnego algorytmu nie będziemy tu przedstawiać, jak chcecie to zajrzycie do kodu źródłowego gry.

Odrodzenie się

Czołg po zniszczeniu znika z mapy i zostanie na nią zrzucony po 55 (heavy) lub 45 (light) tickach.

Akcja czołgu ruchu do wskazanego pola

Czołg może wysłać akcję (GoTo) wskazując x i y pola do którego ma czołg jechać. Pierwsze pole w lewym górnym rogu mapy to punkt (0, 0). Akcja GoTo wykonuje tylko pojedynczy ruch (obrót lub poruszenie się), który przybliży czołg do miejsca docelowego w danym ticku, więc aby faktycznie dotrzeć do wskazanego pola należy wysyłać tę akcję co tick, do momentu aż czołg dotrze do wskazanego miejsca.

Ponadto ta akcja posiada kilka parametrów. Serwer oblicza optymalną ścieżkę na podstawie dostarczonych kosztów i kar, a następnie wykonuje pierwszy krok tej ścieżki. Parametry `GoTo`:

- `x`: Docelowa współrzędna `x` na mapie.
- `y`: Docelowa współrzędna `y` na mapie.
- `turret_rotation`: Opcjonalny obrót wieżyczki. Ten obrót jest wykonywany tylko wtedy, gdy algorytm wyszukiwania ścieżki zdecyduje, że najlepszym natychmiastowym ruchem jest obrót korpusu czołgu. Pozwala to na jednoczesny obrót czołgu i wieżyczki.
- `costs`: Opcjonalne koszty związane z podstawowymi akcjami ruchu (do przodu, do tyłu, obrót).
 - `forward`: Koszt ruchu do przodu o jedno pole.

Hack⚡Arena 2.5

- ``backward``: Koszt ruchu do tyłu o jedno pole.
- ``rotate``: Koszt obrotu czołgu.
- ``penalties``: Opcjonalne kary związane z niebezpiecznymi ruchami lub polami.
 - ``blindly``: Kara za ruch do tyłu nie widząc co na tym polu jest (np. ruch do tyłu, gdy wieżyczka jest skierowana do przodu).
 - ``bullet``: Kara za wjechanie na pole zajęte przez pocisk.
 - ``mine``: Kara za wjechanie na pole zajęte przez minę.
 - ``laser``: Kara za wjechanie na pole zajęte przez promień lasera lub potencjalnie inne własne jednostki.
 - ``tank``: Kara za wjechanie na pole zajęte przez inny czołg.
 - ``per_tile``: Lista określonych pól z dodatkowymi karami. Przydatne w niestandardowych scenariuszach, np. do cachowania znanych lokalizacji min. Każdy element listy (``PerTilePenalty``) zawiera:
 - ``x``: Współrzędna ``x`` pola.
 - ``y``: Współrzędna ``y`` pola.
 - ``penalty``: Dodatkowy koszt kary za wjechanie na to pole.

Hack⚡Arena 2.5

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** (to znaczy nie zbuduje się obraz dockera, nie uruchomi się, będzie się crashował lub poprzez absolutnie tragiczną wydajność spowalniał cały turniej), 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/HackArena2.5-StereoTanks>, 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.

Hack⚡Arena 2.5

Dostępne języki programowania

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

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.

Serwer

Dla każdej drużyny przygotowaliśmy specjalne konto na serwerze, dzięki czemu macie serwer o publicznym IP, do którego będziecie mogli się podłączyć swoimi botami. Aby go włączyć należy zalogować się swoim loginem oraz hasłem podanym na stronie hackarena.pl w panelu użytkownika. Logowanie odbywa się za pomocą terminala na waszych komputerach, wykonujecie polecenie ``ssh login@game.hackarena.pl`` i wpisujecie hasło. Serwer automatycznie wystartuje. Możecie włączyć tylko jeden taki serwer. Aby go zrestartować należy zatrzymać serwer za pomocą CTRL + C i ponownie

Hack⚡Arena 2.5

się zalogować za pomocą SSH. Aby się połączyć teraz np. klientem GUI do serwera, wystarczy przy dołączaniu wpisać w pole host adres game.hackarena.pl:port, port jest wyświetlany po zalogowaniu się na serwer za pomocą SSH (to co robiliście wcześniej). A jak łączyć się botem to podajecie parametry konsolowe `--host game.hackarena.pl --port port``

Hack⚡Arena 2.5

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. Napisaaliśmy skrypt w języku Python wraz z biblioteką aiodocker, 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 9 rund (Best of 9). Pozwoli nam to sprawdzić, który z Waszych botów jest faktycznie najlepszy.

Hack⚡Arena 2.5

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.

Hack⚡Arena 2.5

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.

Hack⚡Arena 2.5

System oceniania

Turniej finałowy HackArena 2.5 zostanie przesymulowany automatycznie.

Aby zagwarantować jak najbardziej rzetelne wyniki, każdy mecz zostanie rozegrany w 9 rundach (Best of 9), czyli 2 drużyny grają 9 powtórek meczu na różnych mapach (różne seedy mapy) i ta drużyna, która wygra 5 rund, wygrywa cały mecz. Więc mówiąc **"mecz"** mamy na myśli dziewięciokrotne przesymulowane rozgrywki, a ta drużyna która wygra większość rund wygrywa. Taka pojedyncza symulacja to właśnie **"runda"** w meczu. Każda runda będzie toczyć się na mapie o wymiarach 20x20, będzie trwać 2400 ticków, broadcast interval wynosić będzie 100 milisekund oraz będzie włączona flaga `--eager-broadcast` na serwerze w celu potencjalnie szybszej symulacji.

Przypadki szczególne

1. Remisy

W przypadku, gdy obie drużyny będą miały taki sam wynik w rundzie oraz tyle samo eliminacji (mało prawdopodobne, ale jednak), następuje po prostu remis i nie w tej rundzie zwycięzcy.

2. Obie drużyny ex aequo w meczu po wszystkich rundach

To może się zdarzyć w przypadku remisów. W takim przypadku, rozgrywamy dodatkową rundę i zwycięzca tej rundy jest wygranym meczu.

3. Mecz z jedną drużyną bez drugiej

Taka sytuacja może nastąpić w przypadku gdy któraś drużyna zostanie zdyskwalifikowana lub ktoś po prostu nie stanie do walki, mecz nie jest rozegrany. I bot automatycznie awansuje do następnego etapu.

Hack⚡Arena 2.5

Powodzenia!

W przypadku jakichkolwiek problemów zgłaszajcie się do organizatorów HackArena 2.5 - z pewnością pomożemy!

Życzymy dobrej zabawy i zwycięstwa w drugiej i pół edycji hackathonu HackArena!

Zespół HackArena
Członkowie KN init