Testing Cup 2015 – Olimpiada FP

Uratuj Deadline24

# Intro

Jest rok 2020. Dzień kolejnej edycji ogólnoświatowego konkursu Deadline24. Z okazji jubileuszu postanowiono wrócić do kopalni i kilku zadań sprzed lat. Tysiące uczestników zjeżdża do kopalni, aby zostać zamkniętych na 24 godziny. Marcin po raz ostatni odpala serwery testowe … komunikacja zostaje przerwana, procesy się zatrzymują, na ekranie konsoli pojawia się:



Marcin wybucha gromkim śmiechem. Próbuje uruchomić usługi jeszcze raz.

It`s not a joke!

by \_gruszkow

Marcin nie wierzy. Przypomina sobie jak przed laty niejaki Dimiriej Gruszkow został zdyskwalifikowany za używanie dysku SSD, technologii wtedy jeszcze niedozwolonej, dzięki której algorytmy wykonywały się o rząd wielkości szybciej. Rosjanie grozili nawet zakręceniem gazu i wykupieniem wszystkich jabłek, ale na szczęście udało się dojść do porozumienia na drodze dyplomatycznej (tylko nieliczni wiedzą o tym, że kosztowało to nas gegapetolitry polskiej wódki). Oto i on! Wrócił, aby popsuć konkurs.

Hi Bro!

I`ve modified a code one of tasks to improve a challenge. I hope you will like it ☺ Have FUN!

Regards

\_gruszkow

Marcin nadal podśmiechuje się … nerwowo. „Pff, co ten gruszkow sobie myśli, że tu pracują ludzie cumSoftu? Zaraz, odpalimy unit testy i fixy zajmą kilka chwil.”

„Zaraz, zaraz … ten kodzik był pisany dawno … nie mamy testów? Jak to? @##$$%&^&^%\*^&\*( MAĆ! Dawaj tu naszych testerów! A, zresztą, dawaj tu wszystkie nasze interdyscyplinarne zespoły!”

Dzięki obiecanego team Spiritu, po 15 minutach stawiają się pierwsze ekipy … ☺

„Marcin, nie martw się, roz#%%$&^& iemy to!”



Quality Excites?

Uratuj Deadline24 i nie pozwól uwięzić tysiące ludzi w kopalni …

# Reguły gry

Konkurs jest podzielony na dwie główne części:

- testowanie zadania „Osada” i serwerów deadline24

- rozwiązywanie zadań (pytania otwarte/zamknięte)

Punktacja za pytania z quizu stanowi ~połowę punktów do zdobycia za błędy w Osadzie. Rozwiązywanie zadań quizu może pomóc w zdobyciu wskazówek do zadania głównego. Pewne wskazówki mogą pojawiać się też w trakcie konkursu niezależnie od wyniku drużyny.

Punktacja za konkretne błędy i pytania jest ustalona z góry. Za błąd z „Osady” można zdobyć od 1-10 punktów. Pytania z quizu punktują 1-3, za wyjątkiem pytania otwartego z przeglądu kodu, gdzie będzie punktować każda uwaga/błąd.

**Bonusy punktowe:**

- drużyna, która pierwsza odnajdzie dany błąd/opowie na dane pytanie prawidłowo zdobywa podwojoną ilość punktów za dane zadanie

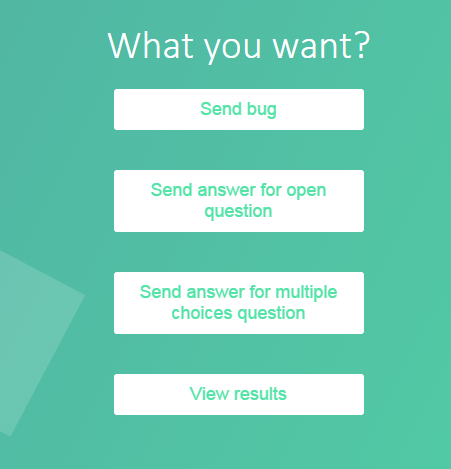
- drużyna, która jako jedyna (co oczywiście implikuje, że jest też pierwsza) znajdzie dany błąd/odpowie na dane pytanie prawidłowo zdobywa poczwórną ilość punktów za dane zadanie

- komisja przyznaje sobie prawo (bo jest komisją ☺) do przyznawanie dodatkowych bonusów za trafne spostrzeżenia/opisy/nietrywialne nieznalezione błędy

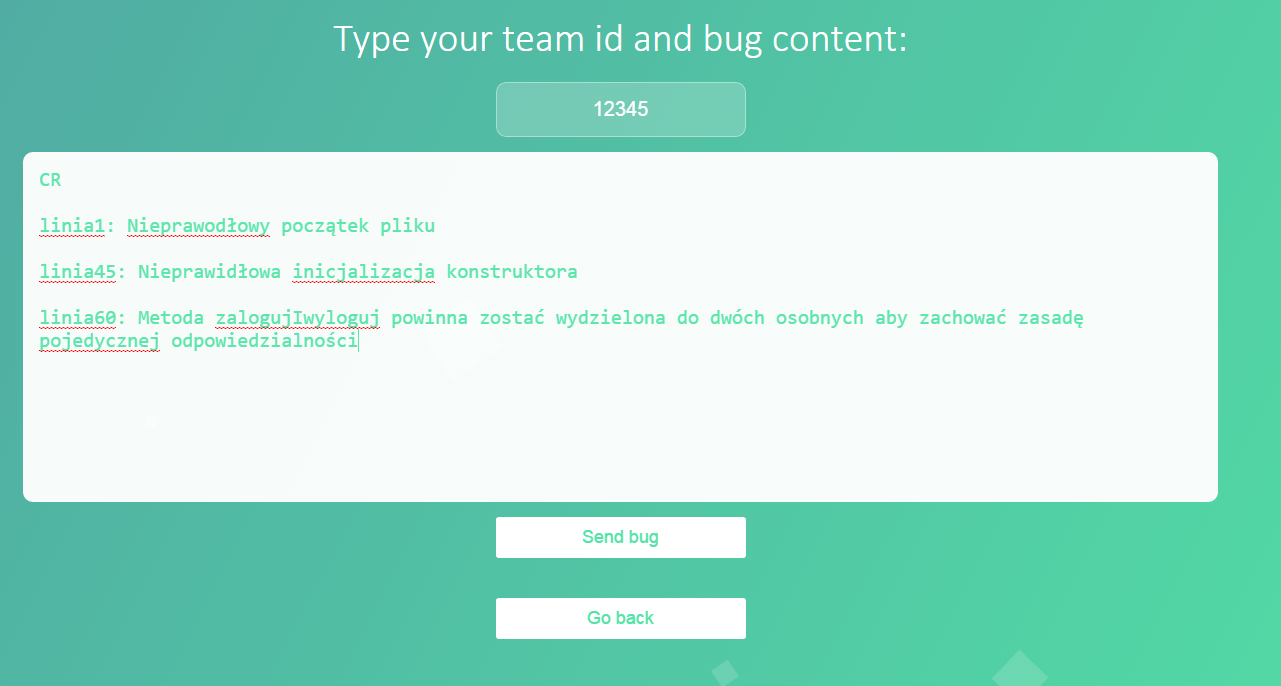
# Zgłaszanie błędów

Znajduje się pod adresem testingcup.fp:5001

Każda drużyna została zasilona w unikalny identyfikator w postaci GUIDa, którym przedstawia się za każdym razem, kiedy chce dodać błąd w systemie, lub odpowiedzieć na pytanie.



1. Send Bug – tutaj zgłaszamy wszystkie błędy z Osady, wszystkie błędy spoza quizu, oraz pytania z Code Review. W przypadku Code Review – uwagi można wysyłać dowolną ilością zapytań, można je zgromadzić w jeden opis, ale też można każdą uwagę posłać osobno. Warto, aby podać w treści, że uwagi dotyczą CR i odnosić się do konkretnych linii lub zakresu np.



1. Send answer for open question – dotyczy pytań otwartych (to te, które nie mają odpowiedzi do wyboru). Aby odpowiedzieć, trzeba podać ID pytania.
2. Send answer for multiple choices question – odpowiadamy na pytania zamknięte, pytania wielokrotnego wyboru. Aby odpowiedzieć, trzeba podać ID pytania.
3. View results – aktualne wyniki poglądowe (sondażowe ;))

**UWAGA:** Próby spamowania sewera czy innego hackowania grożą dyskwalifikacją. Serwer zgłaszania błędów nie jest przedmiotem testowania i służy sprawniejszemu przeprowadzeniu konkursu.

# Quiz

Pytania quizowe znajdują się w pliku testingCupQuiz.pdf

# Co testujemy?

* Główne funkcjonalność i zasady gry „Osada”
  + Zgodność z wymaganiami – z opisem rozgrywki
  + API – commandy, zapytania i odpowiedzi
  + Naliczanie punktów
  + Zachowanie obiektów na planszy, zasady poruszania itp.
  + Konfiguracja serwerów głównie powinna się przydać, jako pomoc w testowaniu
* Nie testujemy
  + Serwera www (logowanie, UX, bezpieczeństwo)
  + Jak serwery reagują na błędną konfigurację – chyba, że będzie to uzasadnione
  + Wydajności

# Maszyna wirtualna

System do testowania jest udostępniony w formie obrazu VirtualBox. Aby uruchomić maszynę należy dostosować ustawienia do swoich potrzeb i swojego komputera (liczba rdzeni procesora, ilość pamięci RAM, kontroler sieci, kontroler USB) a następnie kliknąć przycisk ‘Start’.

Na dostarczonej maszynie zainstalowane jest Ubuntu.

Najwygodniej logować się do maszyny za pomocą ssh (putty).  
Dane do logowania to dl24/d24, na tym użytkowniku należy wykonywać wszystkie akcje.  
Maszyna jest (powinna być) skonfigurowana dla zorganizowania konkursu dla 30 drużyn.

Drużyny posiadają loginy i hasła. Kolejno od *team1/pass1* do *team30/pass30.* Tych loginów i haseł używamy do połączenia z serwerami zadań i logowania do aplikacji www.

**Waszym zadaniem jest sprawdzenie, czy maszyna jest gotowa na konkurs. Czy wszystko jest odpowiednio skonfigurowane, a zadanie (Osada) działa zgodnie z dostarczonym opisem.**

## Pomocne komendy Unix

### Przeglądanie logów – *less*

Skróty klawiszowe:

* *SHIFT+G* – idź na koniec
* */pattern* – wyszukanie wzorca w pliku
  + *n* – znajdź następny
  + *SHIFT + n* – znajdź poprzedni

### Edycja plików – *nano*

Skróty klawiszowe:

* *CTRL+o* –zapis
* *CTRL+x* –wyjście

## Komunikacja z serwerem zadań – netcat

Przykładowa komunikacja znajduje się w opisie zadania.

- ustawić poprawnie konfigurację startu gry w konfigach

# Przydatne informacje o konfiguracji maszyny dl24

Pliki serwera dl24 znajdują się w katalogu */home/dl24/contest/;*

## global-server

Jest komponentem koniecznym do działania aplikacji, odpalamy go poleceniem *dl24-global-server* w katalogu */home/dl24/contest/global*

## serwer www

Zawiera aktualną punktację zawodów, możliwość logowania na loginy takie jak dla serwerów z zadaniami. Dostęp do serwera (aplikacji z wynikami) na domyślnym porcie 80 na dostarczonej maszynie. (localhost)

# Serwery z zadaniami

### Uruchamianie

Pierwszy krok to ustawienie daty startu konkursu w pliku settlement\_p.conf (parametr start\_time). W momencie startu (a właściwie po wykonaniu przez jednego z graczy komendy innej niż zalogowanie) serwer symuluje grę od daty startu do daty aktualnej. W związku z tym w celu usprawnienia testów zaleca się ustawienie daty bliskiej aktualnej.

Uruchamiamy komendą *dl24-settlement-server*, w jednym z folderów który oznacza kolejny serwer, Osada1, Osada2, Osada3

* */home/dl24/contest/s1*
* */home/dl24/contest/s2*
* */home/dl24/contest/s3*

W trakcie działania serwera, co jakiś czas jego stan jest zapisywana do podfolderu *storage*. W przypadku, gdy chcemy serwer uruchomić od początku (a zwłaszcza, gdy zmieniliśmy konfigurację) należy usunąć zawartość tego folderu.

Przy uruchomieniu w sposób podany powyżej logi maszyny lądują na konsolę. Istnieje możliwość przekierowania logów do pliku dodając do komendy uruchomienia *>> nazwa\_pliku*. Logi mogą być pomocne przy analizie poprawności działania aplikacji.

# Konfiguracja

W folderze każdego z serwerów znajdują się pliki konfiguracyjne. Domyślnie powinny być one skonfigurowane zgodnie z treścią zadania. W celu ułatwienia testów możliwa jest edycja tych plików.

Pliki konfiguracyjne:

## settlement\_p.conf

port – port serwera

start\_time – czas startu (patrz uruchamianie)

## settlement.conf

turnDuration - czas trwania tury w sekundach

turnsCount - plik z ilością tur dla kolejnych rozgrywek

terrainDesc - plik z nazwami plików z mapami dla kolejnych rozgrywek

standardsDesc - plik z nazwami plików z położeniem sztandarów dla kolejnych rozgrywek

defendersDesc - plik z nazwami plików z ilością obrońców dla kolejnych rozgrywek

immigrationsDesc - plik z nazwami plików z opisem kolejnych fal ataków coor

mapRemains - plik opisujący czy mapa zostaje na kolejną rozgrywkę

commandsLimitPerTurn - limit komend na turę dla każdej drużyny

gamesCount - liczba rozgrywek

## plik z ilością tur dla kolejnych rozgrywek - domyślnie times.desc

w każdej linii znajduje się czas trwania kolejnej rozgrywki w turach

## plik z nazwami plików z mapami dla kolejnych rozgrywek – domyślnie maps.desc

w kolejnych liniach ścieżki do plików z mapą dla kolejnych rozgrywek

### opis pliku z mapą – domyślnie pliki w katalogu maps

W pierwszej linii 2 liczby oddzielone spacją – wymiary mapy – mapa powinna być kwadratowa.

Następnie w kolejnych liniach opis kolejnych pól planszy:

* 0 – teren stabilny
* 1 – teren niedostępny
* 2 – teren grząski
* 3 – miejsce na maszt gracza, liczba masztów musi być równa liczbie graczy (30)

Powyższy opis staje się bardziej zrozumiały po obejrzeniu plików na maszynie.

## plik z nazwami plików z położeniem sztandarów dla kolejnych rozgrywek – domyślnie standards.desc

w kolejnych liniach ścieżki do plików z położeniem sztandarów dla kolejnych rozgrywek

### opis pliku z położeniem sztandarów – domyślnie pliki w katalogu standards

W pierwszej linii liczba pól (30). W kolejnych liniach liczba porządkowa i współrzędne kolejnego punktu – współrzędne liczone od 0.

Podane współrzędne masztów muszą zgadzać się z tymi z mapy.

## plik z nazwami plików z ilością obrońców dla kolejnych rozgrywek – domyślnie defenders.desc

w kolejnych liniach ścieżki do plików z ilością obrońców dla kolejnych rozgrywek

### opis pliku z ilością obrońców – domyślnie pliki w katalogu defenders

W jednej linii kolejno:

* Liczba szturmowców
* Liczba snajperów
* Liczba inżynierów

## plik z nazwami plików z opisem kolejnych fal ataków coor – domyślnie immigrations.desc

w kolejnych liniach ścieżki do plików z opisem kolejnych fal ataków coor

### Opis pliku z opisem fal ataków – domyślnie pliki z katalogu immigrations

W pierwszej linii – liczba fal ataków. W kolejnych liniach opis kolejnych ataków:

* Czas wystąpienia
* Liczba siepaczy
* Liczba złodziei
* Liczba maruderów

## plik opisujący czy mapa zostaje na kolejną rozgrywkę – domyślnie map\_remains.desc

w kolejnych linia 1 – mapa zostaje, lub 0 – mapa się zmienia