

PROGRAMOWANIE OBIEKTOWE - PROJEKT

OGÓLNY OPIS PROJEKTU

Celem projektu jest stworzenie symulatora świata superbohaterów. Podstawowymi bytami w tym świecie powinny być *miasta*, *cywile*, *superbohaterowie*, *złoczyńcy* i *źródła mocy*.

Najprostszymi bytami w symulowanym świecie są źródła mocy. Każde źródło mocy powinno posiadać unikatowy identyfikator, typ/nazwę oraz potencjał. Ponadto, każdy typ źródła powinien mieć ściśle określoną zdolność, którą wzmacnia (zdolności zostaną opisane w późniejszych akapitach). Przykładowymi źródłami mocy w symulowanym świecie mogą być np. czerwony kryptonit, który zwiększa siłę lub niebieski kamień nieskończoności, który zwiększa inteligencję.

Źródła mocy są ukryte w miastach. Każde miasto posiada nazwę, liczbę mieszkańców (cywilów), oraz typy źródeł mocy. Każde miasto może posiadać więcej niż jeden typ źródła mocy, lecz żadne miasto nie może posiadać wszystkich typów źródeł.

W miastach mieszkają cywile. Każdy cywil posiada unikalne imię i nazwisko oraz miasto rodzinne. Liczba mieszkańców przebywających w danym mieście wpływa na prędkość zwiększania się potencjału źródeł mocy znajdujących się w tym mieście. Co pewien czas potencjał każdego źródła mocy zwiększa się proporcjonalnie do liczby mieszkańców miasta. Tym samym źródła w dużych miastach będą szybciej zwiększały swój potencjał. Dodatkowo, co jakiś czas każdy cywil losowo wybiera miasto, do którego ma iść. Po jego odwiedzeniu wraca do miasta rodzinnego, a następnie losuje następne miasto do odwiedzenia. Pomiedzy miastami, cywile poruszają się specjalnie wyznaczonymi drogami. Jeśli miasto rodzinne zostanie zniszczone pod nieobecność cywila, miasto w którym się znajduje (lub które jako ostatnie odwiedził) staje się jego nowym miastem rodzinnym.

Co pewien czas na miasta napadają złoczyńcy. Każdy złoczyńca posiada imię, poziom życia oraz w różnym stopniu rozwinięte sześć zdolności: inteligencję, siłę, szybkość, wytrzymałość, energię oraz umiejętność walki. Po pojawieniu się na jednym z krańców symulowanego świata, złoczyńca wybiera najbliższe miasto i na nie napada. By dostać się do miasta, złoczyńcy korzystają z dróg, na których zabijają każdego napotkanego cywila (niezależnie od tego czy podróżował w tym samym czy przeciwnym kierunku). Gdy złoczyńca dotrze do miasta, powoli pochłania potencjał zebrany w miejscowych źródłach mocy i stopniowo zmniejsza liczbę mieszkańców. Pochłaniając potencjał, złoczyńca zwiększa swoje zdolności, odpowiednio do pochłanianych źródeł mocy. Gdy liczba mieszkańców i potencjały wszystkich źródeł spadną do zera, złoczyńca losowo wybiera następną osadę do zaatakowania.

Gdy złoczyńcy pojawiają się na świecie, stolica (wyszczególnione miasto) może wysłać na pomoc superbohaterów. Liczba superbohaterów, których można wysłać zależy od łącznego potencjału źródeł mocy we wszystkich miastach. Każdy superbohater posiada imię, poziom życia oraz w różnym stopniu rozwinięte sześć zdolności: inteligencję, siłę, szybkość, wytrzymałość, energię oraz umiejętność walki. Gdy superbohater spotka na drodze (lub w mieście) złoczyńcę, dochodzi do starcia. Starcie powinno polegać na turowym zadawaniu ciosów opartych na zdolnościach. Pierwszy cios powinna zadać postać, która ma lepiej rozwiniętą szybkość. Cios bierze pod uwagę jedną z trzech zdolności – inteligencję, siłę lub energię - i mnoży ją razy umiejętność walki. Od takiej wartości odejmowana jest wytrzymałość osoby przyjmującej cios, by określić ile straciła punktów życia. Gdy poziom życia spadnie do zera, złoczyńca/superbohater umiera. Po ewentualnym zwycięstwie superbohater wraca do stolicy.

WYMAGANIA

DOSTĘPNE FUNKCJE I PODSTAWOWE ZAŁOŻENIA:

- użytkownik tworzy nowych cywilów i superbohaterów poprzez **panel kontrolny**;
- superbohaterowie i cywile są tworzeni tylko na polecenie użytkownika, złoczyńcy pojawiają się automatycznie w losowych odstępach czasu;
- stworzony cywil/superbohater powinien mieć automatycznie wylosowane wszystkie wartości pól;
- wszyscy cywile, złoczyńcy i superbohaterowie są **rysowani na mapie świata** zgodnie z ich aktualnym położeniem;
- każdy cywil, superbohater, złoczyńca to **osobny wątek**;
- użytkownik może oglądać podstawowe informacje o obiekcie (cywila, mieście, złoczyńcy, superbohaterze) w osobnym **oknie lub panelu informacyjnym** po kliknięciu na narysowany na mapie obiekt;
- korzystając z przycisków w oknie informacyjnym obiektu użytkownik może **usunąć** cywila z mapy, **zmienić** następną osadę docelową cywila, zmusić cywila do **awaryjnego postoju** i tym samym utworzyć korek na drodze, **wznówić podróż** cywila;
- cywil może się zatrzymać na drodze zarówno **losowo** jak i na polecenie użytkownika;
- aby usunąć korek na drodze, użytkownik musi usunąć cywila z mapy lub skorzystać z przycisku wznów podróż w oknie informacyjnym;
- na mapie powinno znajdować się **minimum 10 miast**, w tym wyszczególniona **stolica**
- każde miasto powinna posiadać co najmniej jedno źródło mocy; żadne miasto nie powinno mieć wszystkich typów źródeł mocy;
- świat powinien posiadać co najmniej **6 różnych typów źródeł mocy**;
- drogi między miastami powinny się krzyżować ze sobą; każda droga między dwoma miastami powinna posiadać co najmniej jedno skrzyżowanie; na **skrzyżowaniu** dróg może się w danej chwili znajdować tylko jeden cywil; do zapewnienia bezpieczeństwa na skrzyżowaniu należy wykorzystać mechanizm *semaforów* lub *monitorów*;
- wszystkie drogi powinny być **dwukierunkowe**; oba pasy ruchu każdej drogi powinny być graficznie rozdzielone tak aby cywile jadący w przeciwnych kierunkach nie najeżdżali na siebie;
- aplikacja nie ma rozwiązywać żadnych problemów zakleszczeń; jeśli na jakimś skrzyżowaniu wystąpi zakleszczenie, jego rozwiązanie będzie możliwe poprzez usunięcie cywila z drogi;
- modelowany świat powinien być bezpieczny – **żadnych kolizji ani nieudanych postojów**;
- po zakończeniu symulacji poprzez zmniejszenie się liczby ludności i sumy potencjałów wszystkich miast do zera, aplikacja powinna poprosić o imię użytkownika i zapisać w osobnym pliku czas gry;
- w pliku powinno być zapisywanych tylko **5 najlepszych wyników w kolejności malejącej**;
- plik z wynikami powinien być tworzony poprzez serializację do XML'a;
- podstawowymi kryteriami oceny są kod i funkcjonalność zgodna z wymaganiami; walory estetyczne wizualizacji działają na plus oddającego, ale brak wymyślnej grafiki w żaden sposób nie obniża oceny;
- podobnie jak walory estetyczne, choć niewymagane, na plus będą działać nawiązania do kultury popularnej (np. Batman, Spiderman, Avengers, Justice League, X-Men).

DODATKOWE WYMAGANIA:

- w terminie najpóźniej do 19. października 2014 należy przesłać lub osobiście przedstawić prowadzącemu swoją wizję modelowanego świata w postaci **diagramu klas UML**;
- w terminie najpóźniej do 26. października 2014 należy przedstawić prowadzącemu **klasy Java** napisane zgodnie z zaproponowanym diagramem klas;

- wszystkie pola klas powinny być prywatne (lub ewentualnie chronione), a dostęp do nich powinien być realizowany poprzez **settery** i **getterzy**;
- kod projektu powinien być udokumentowany tak by umożliwić wygenerowanie dokumentacji w formacie HTML przy pomocy narzędzia **javadoc**;
- nadesłany projekt powinien zawierać aplikację w postaci **uruchamialnego** pliku *.jar, folder ze źródłami oraz prosty plik *readme* zawierający:
 - imię, nazwisko, numer indeksu, grupę i dzień zajęć,
 - krótką instrukcję obsługi programu;
- kod źródłowy programu wraz z dodatkowymi plikami powinien zostać przesłany na adres prowadzącego **do 09.01.2015 do 23:59**; po tym terminie ocena będzie co tydzień obniżana o ocenę w dół lub wymagane będzie rozbudowanie projektu;
- zastrzega się możliwość wezwania studenta w celu osobistego wyjaśnienia wątpliwości dotyczących programu zaliczeniowego;
- **wszelkie plagiaty będą skutkowały oceną 2,0 zarówno dla osoby udostępniającej jak i kopiującej kod.**