

Gra Koronawirus AntiPlague

Napisz program będący odwrotnością gry w stylu "Plague Inc.". Podczas rozgrywki zaprezentowana będzie mapa świata w podziale na Państwa (wybierz minimum 10 dowolnych Państw do realizacji). Na mapie tej zrealizowane różne sposoby podróży pomiędzy krajami (linie lotnicze, autokary, pociągi, pojazdy samochodowe, cargo itp.). Sposoby podróży są dowolne, jednak musi być ich co najmniej 3 (nie wszystkie muszą być pomiędzy wszystkimi krajami, jednak z każdego kraju co najmniej 2 sposoby transportu). Gra polega na przeciwdziałaniu zarażenia wszystkich osób we wszystkich Państwach wirusem. Nie ma obowiązku odwzorowywania rzeczywistych Państw - jeśli ktoś chce umieścić rozgrywkę np. w jakimś innym uniwersum, to jest taka możliwość.

Każde Państwo posiada inne kryteria wyłączenia każdej z dróg transportu pomiędzy krajami (kryteria wg. własnego uznania - np. wg. liczby zarażonych). Każde z kryteriów musi być możliwe do osiągnięcia i modyfikacji za pomocą wykupywanych ulepszeń. Dobór ulepszeń i ich działanie pozostawiam Państwa inwencji twórczej. Ulepszeń musi być co najmniej 9.

Ulepszenia można kupować za punkty zbierane za każdą osobę, która została uratowana przed zarażeniem lub wyleczona. Wirus zaczyna swoją drogę w wybranym państwie na początku programu (losowo lub przez użytkownika). Sposób, prędkość i efekty rozprzestrzeniania się są zależne od poziomu trudności i zaimplementowane według Państwa uznania.

Gra kończy się w sytuacji, gdy uda nam się uchronić całą społeczność świata i zastopować zarażenia lub gdy wszyscy zostaną zarażeni. Oczywiście należy zaimplementować również niezależnie postępujące zarażenia wirusem i rozprzestrzenianie się go na świecie.

Należy zapewnić w pełni funkcjonalny interfejs graficzny. Konsola poleceń (*CLI*) może być jedynie pomocą informacyjną, ale nie może zachodzić tam żadna interakcja użytkownika z programem.

Program po uruchomieniu powinien wyświetlać menu główne składające się z opcji:

- *New Game* - nowa gra
- *High Scores* - tabela wyników
- *Exit* – wyjście

Po uruchomieniu nowej gry, gracz zostanie zapytany w osobnym oknie jaki stopień trudności chce rozegrać (implementacja stopni trudności dowolna). Po uruchomieniu gry w nowym oknie wyświetlana jest plansza gry, a licznik czasu rusza (warto zauważyć, że licznik czasu, podobnie jak postępowanie wirusa i inne powinny być **realizowane w osobnych wątkach**, żeby nie blokować interakcji z oknem). Podczas gry musi być widoczny licznik punktów oraz czasu, aktualizowane na żywo podczas rozgrywki. Połączenia transportowe pomiędzy krajami należy wizualizować wykonując proste animacje (np. dla transportu lotniczego poruszająca się po ścieżce ikona samolotu). W projekcie należy wykorzystać pliki graficzne oraz postarać się o miły dla oka efekt wizualny całej aplikacji z uwzględnieniem spójności wyglądu okien takich jak *Menu*, *Ranking* i *główne okno z rozgrywką*. Gra toczy się według wyżej wymienionych zasad. Należy zapewnić możliwość przerywania gry w dowolnym momencie poprzez **złożony skrót klawiszowy** (*Ctrl+Shift+Q*), który spowoduje powrót do menu głównego. Należy zapewnić, żeby skrót działał w każdym momencie rozgrywki.

Po zakończeniu gry, w nowym oknie gracz proszony jest o swoją nazwę pod jaką ma być zapisywany w rankingu.

Ranking liczony jest na podstawie czasu, uzyskanych efektów i stopnia trudności (dowolny sposób implementacji). Należy zapewnić trwałość rankingu po ponownym uruchomieniu aplikacji, czyli należy go przechowywać w pliku na dysku. Do zapisu informacji o rankingu należy wykorzystać interfejs *Serializable*.

Po wybraniu opcji rankingu z menu głównego, zostaje on wyświetlony użytkownikowi. Ponieważ okno rankingu może być relatywnie duże, dlatego należy zadbać o paski przewijania, w razie gdyby nie mieścił się on w oknie racjonalnych rozmiarów.

Wskazówki:

- Należy zadbać o wyjątki w programie. Jeśli jakiś wystąpi należy wyświetlić jego komunikat użytkownikowi.
- Ranking należy zrealizować przy pomocy komponentu *JList*.
- Należy zadbać o skalowalność okien aplikacji.
- Państwa można realizować przy pomocy przycisków z odpowiednimi grafikami (jednak tak, aby wyglądało to estetycznie), ale można też zaprojektować własny komponent.
- Nie wszystkie okna muszą być realizowane poprzez klasę *JFrame*. Przy mniejszych i informacyjnych oknach można wykorzystać okna dialogowe.

Projekt opiera się o materiał z zakresu GUI.

Należy zastosować w projekcie wzorzec projektowy MVC.

Uwaga:

- *W przypadku otrzymania projektu ze znacznymi brakami w implementacji/z niezaimplementowaną rozgrywką lub niekompilującego się, wynikiem za taki projekt będzie 0 pkt.*
- *Zabrania się wykorzystywania narzędzi WYSIWYG do generowania okien (tzw. Window Builder'ów).*
- *Brak znajomości dowolnej linii kodu lub plagiat skutkować będzie wyzerowaniem punktacji za ten projekt.*
- *W ocenie projektu poza praktyczną i merytoryczną poprawnością będzie brana również pod uwagę optymalność, jakość i czytelność napisanego przez Państwa kodu.*
- *Ważną częścią projektu jest wykorzystanie między innymi: dziedziczenia, kolekcji, interfejsów lub klas abstrakcyjnych, lambda-wyrażeń, typów generycznych, dodatkowych funkcjonalności lub struktur oraz innych elementów charakterystycznych (ale tylko w naturalny sposób, nic na siłę)*