Student: Jacek Janczura (2E3) **Prowadzący**: Dominik Kasprowicz
janczura.jacek@gmail.com

Programowanie Obiektowe (PROE)- projekt 2
Specyfikacja strukturalna (techniczna)

Temat: Walki orków i graficzna reprezentacja stanu obu armii po każdym starciu.

Struktury danych: Program posługuje się klasami, tworzy na ich podstawie obiekty, których parametry wczytywane są z pliku tekstowego.

Sposób działania programu: Program pobiera strumień danych z pliku tekstowego po czym na jego podstawie, a konkretniej na podstawie pierwszego parametru tj. Typu orka tworzy obiekty klas: Ork, OrkAdiusz, OrkNitolog, TopOrk, Orkhybryda, którym przydzielane są parametry wpisane w pliku. Przy tworzeniu orków jednocześnie przydzielane są im losowo numery od 0 do 1 które oznaczają przydział do armii 1 lub armii 2. Armia która ma sumarycznie więcej parametru inteligencja zaczyna walkę, Warunkiem końca bitwy jest inteligencja =0 co oznacza, że w którejś armii nie ma już więcej orków. Orki ustawione są w vectorze. Z armi zaczynającej atak wybierany jest pierwszy ork, który atakuje ostatniego orka z drugiej armii getSize()-1 oczywiście jeżeli takowy ork istnieje, a wskaźnik na nie go nie jest nullptr. Atak polega na odebraniu orkowi atakowanemu tylu punktów życia, ile ataku ma ork atakujący. Jeżeli życie orka jest mniejsze równe 0 ork jest usuwany. Usuwanie przebiega następująco: Jeżeli usuwanym orkiem z vectora jest ostatni ork to zwyczajnie wywoływane jest pop_back() i ork (wskaźnik) jest usuwany, a następnie pamięć jest zwalniana delete ork i w rezultacie następuje usunięcie orka. Jeżeli ork nie jest ostatni na jego miejsce wstawiany jest ork ostatni i usuwany jest ork do usunięcia z ostatniego miejsca wywołaniem pop_back() i zwalniana jest pamięć delete ork. Po skończeniu ataków podliczana jest sumaryczna inteligencja obu armii i jeżeli nie równa się 0 przechodzimy do analogicznej 2 walki i tak aż do spełnienia warunku końca. Po każdej walce tworzony jest plik html z grafikami SVG przedstawiającymi stan obu armii. Przed pierwszą bitwą w pliku index.html jest pokazany stan przed bitwą i odnośnik do pierwszej pitwy. Po pierwszej bitwie pokazany jest stan po bitwie i tworzone są 2 odnośniki do pliku sprzed bitwy i następnego. W przypadku gdy jest to końcowa walka, odnośnik do następnej nie jest wyświetlany. Zatem po bitwach mamy wgląd w stan obu armii, a do tego możemy przejść do stanu sprzed bitwy i po bitwie.

Klasą podstawową jest sztucznie stworzona klasa Istota po której dziedziczą pozostałe orki. Jest to klasa czysto wirtualna i służy jako „stelaż dla reszty orków", została utworzona tylko dlatego żeby później wygodnie można było stworzyć dziedziczenie wirtualne, wielodziedziczenie bez przerabiania całej klasy Ork. Klasa ork Hybryda jest klasą dziedziczącą jednocześnie po klasie OrkNitolog jak i po klasie TopOrk i została rozszerzona

dodatkowo o parametry specjalne tych 2. klas. Klasy OrkNitolog i TopOrk dziedziczą wirtualnie po klasie Ork żeby w klasie Orkhybryda wyeliminować duplikacje metod, parametrów etc. Przeciążony został parametr <<, w sumie służy on spełnieniu jednego z poleceń zamieszczonych w opisie projektu. Zmiana dotyczy jedynie wyświetlania informacji o orkach zamiast:

```
this->getInfo ();

drugiOrk->getInfo ();
```

Używamy cout<< któey „wypluwa” informacje z get info przez operator <<

```
cout<<this->getInfo ();

cout<<drugiOrk->getInfo ();
```

Zmiany w get info: Na początku

```
void Ork::getInfo ()

{cout << "\t"<< nazwa << "\t" << atak << "\t" << zycie << "\t" << szybkosc<< "\t" << inteligencja << endl;

}
```

Z voida zmieniłem na stringa i zamiast odrazu cout…, nowe get info tylko zwraca informacje o orkach

```
string Ork::getInfo ()

{       return "\t"+ nazwa + "\t" + to_string(atak) + "\t" + to_string(zycie) + "\t" + to_string(szybkosc)    + "\t" + to_string(inteligencja) +"\n";

}
```

Format danych wejściowych: dokument tekstowy przechowujący dane o orkach

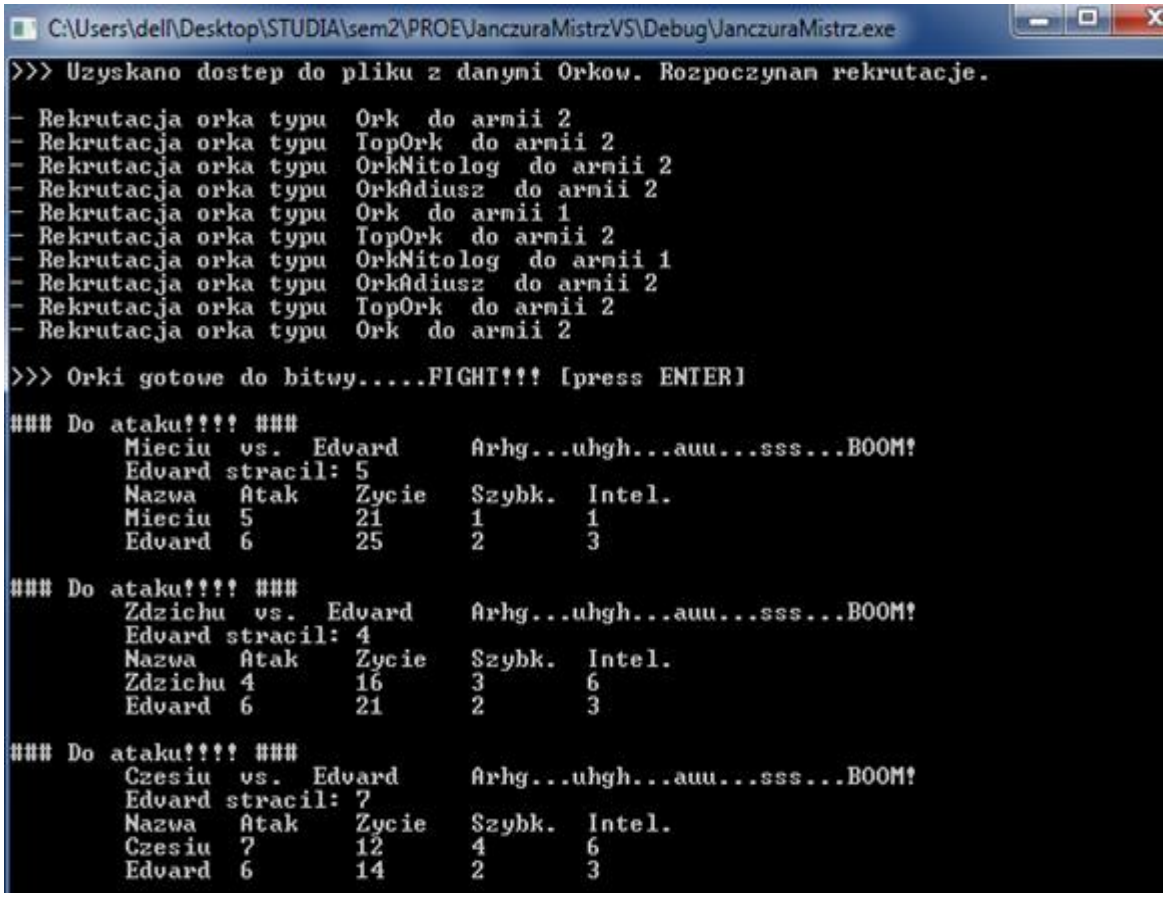
Format danych wyjściowych: Pliki html przedstawiające stany armii po bitwach i przed.

Podział na pliki: Kod podzielony jest na pliki:

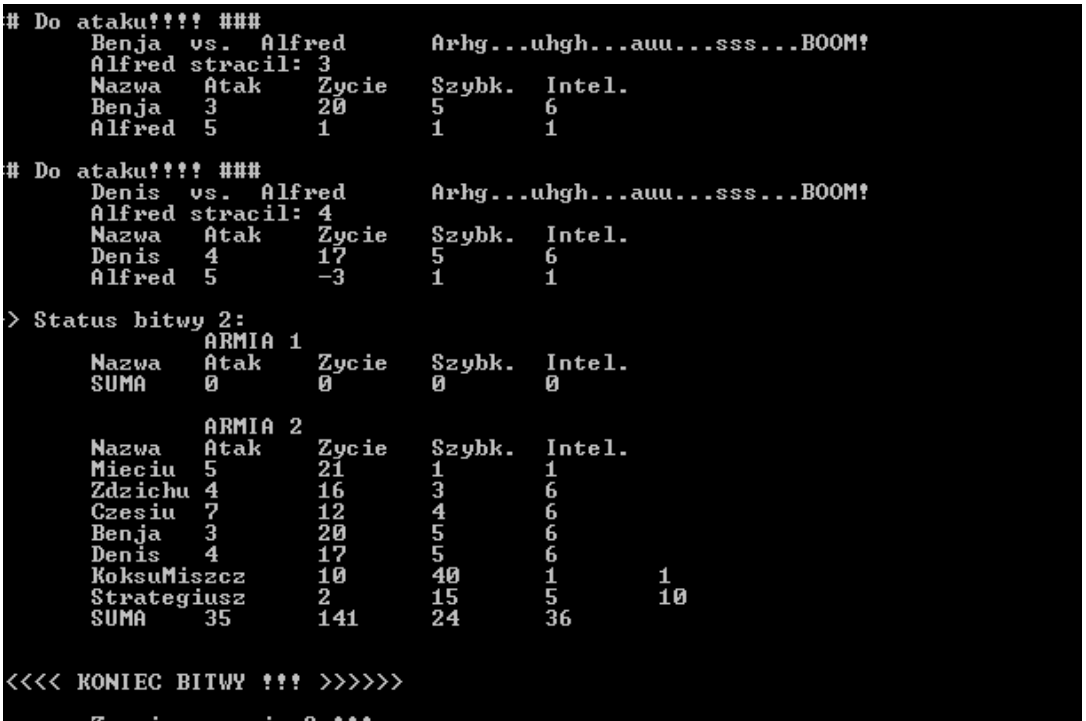
1. Main.cpp
2. Aplikacja.cpp i Aplikacja.h
3. Dokument.cpp i Dokument.h
4. Ork.cpp i Ork.h, Istota.h
5. OrkNitolog.cpp, OrkAdiusz.cpp, TopOrk.cpp, Orkhybryda.cpp, OrkNitolog.h, OrkAdiusz.h, TopOrk.h, Orkhybryda.h,
6. daneOrkow.txt

7. SVG_ork.html SVG_orkadiusz.html SVG_orknitolog.html SVG_topork.html
8. Folder bitwy w którym zapisywane są stany obu armii

Przykładowy wynik działania programu:



……



Screen z działania programu: