

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Zaawansowane programowanie w C++

Dokumentacja wstępna

Kamil Gabryjelski, Antoni Róžański

Prowadzący: Konrad Grochowski

Warszawa, 2017

# 1. Temat projektu

Tematem projektu jest gra Core Wars. Dwóch graczy pisze programy (tak zwanych "wojowników") w języku podobnym do assemblera, które rywalizują między sobą o kontrolę zasobów wirtualnej maszyny. Wygrywa ten gracz, którego procesy zajmą całą pamięć lub wyeliminują wszystkie procesy przeciwnika. Pojedynek toczony jest w specjalnym symulatorze, który dba o kolejność wykonywania instrukcji i czuwa nad ich poprawnością.

## 2. Funkcjonalność

Nasz projekt będzie oferował użytkownikom następujące funkcjonalności:

### 2.1. Instrukcje RedCode

Językiem, w którym gracze będą pisali swoje programy, będzie Redcode. Zestaw instrukcji, które zamierzamy zaimplementować, jest zgodny ze standardem *ICWS '88* i zawiera polecenia: *MOV*, *ADD*, *SUB*, *JMP*, *JMZ*, *JMN*, *CMP*, *SLT*, *DJN*, *SPL*, *NOP*. Jeśli postępy w pracy nad projektem będą zadowalające, graczom udostępnione zostaną dodatkowe instrukcje wraz z modyfikatorami, opisane w dokumencie *ICWS '94 Standard Draft*.

### 2.2. Tworzenie wojowników (Wprowadzanie instrukcji)

Każdy wojownik (program w języku RedCode) będzie się składał z zestawu instrukcji, opisanych w punkcie 2.1. Gracze będą mieli możliwość wprowadzenia tych instrukcji do programu na kilka sposobów:

1. Wprowadzenie instrukcji z poziomu interfejsu graficznego:  
Gracz dostanie możliwość tworzenia programu poprzez zaznaczanie widocznych na ekranie elementów wizualnych, które pozwolą na wstawianie instrukcji w pożądaną kolejność.
2. Wczytanie kodu RedCode z pliku tekstowego:  
W początkowych iteracjach projektu założymy, że program jest napisany poprawnie (bez błędów składniowych). W późniejszych etapach może zostać dodana walidacja wczytywanych plików.
3. Pobranie kodu z bazy danych, np. NoSQL:  
Gdy bazowa funkcjonalność zostanie zaimplementowana, oraz postanowimy, że taka funkcjonalność będzie pożądana w naszej aplikacji, zostanie dodana także integracja z bazą danych, dzięki czemu możliwe będzie zapisywanie i wczytywanie utworzonych wcześniej wojowników.

### 2.3. Interfejs graficzny

Planujemy udostępnić użytkownikowi interfejs graficzny z poziomu przeglądarki internetowej. Za jego pomocą będzie odbywać się wszelka interakcja użytkownika z grą - tworzenie wojowni-

ka, a następnie wizualizacja pojedynku a także prezentację interesujących danych dotyczących przebiegu gry.

Zamierzamy zwizualizować stan rywalizacji jako dwuwymiarową tablicę, której każda komórka odpowiadać będzie adresowi w pamięci. Kolory tych komórek odpowiadać będą procesom użytkowników, co umożliwi ocenę, jaką część pamięci kontroluje każdy z graczy.

### 3. Narzędzia

Narzędzia, z których zamierzamy korzystać:

1. Środowisko programistyczne - JetBrains CLion
2. Kompilatory - Windows: MinGW oraz Linux: GCC
3. Testy jednostkowe - Google Test
4. Kontrola wersji - git/GitHub
5. Optymalizacja kodu - gprof
6. Generowanie dokumentacji - Doxygen
7. Komunikacja - Slack