

---

# „DYNA BLASTER” – PROJEKT PROTOKOŁU SIECIOWEGO

Protokół sieciowy umożliwia komunikację klient – serwer. Zastosujemy protokół tekstowy, w którym porcją przesyłu jest linia tekstu kończąca się znakiem końca linii, parametry dla poszczególnych zdań protokołu, parametry separowane spacjami

## PODSTAWOWE ZAŁOŻENIA

W naszym projekcie zakładamy, że protokół sieciowy będzie odpowiadał za:

- Pobieranie z serwera pliku konfiguracyjnego
- Pobieranie z serwera listy najlepszych wyników (nazwy graczy oraz ich wyniki)
- Pobieranie z serwera plansz poziomów
- Wysyłanie na serwer wyników graczy

Założenia działania serwera:

- Serwer odpowiada na każde żądanie w odpowiedni sposób.
- Wszystkie żądania są typu **String**.
- Wyróżniamy następujące żądania: „GETCONFIG”, „GETMAP id”, „GETSCORES”, „PUSH nick score”. Wszystkie pozostałe żądania są nieprawidłowe.

## REALIZACJA PROTOKOŁU W PROJEKCIE

- Po uruchomieniu gry Klient wysyła do serwera żądanie pobrania pliku konfiguracyjnego:

K: „GETCONFIG” => S

W pliku Config znajdują się takie dane jak: rozmiary okien, ścieżki do plików graficznych czy parametry opisujące rozgrywkę.

- W odpowiedzi serwer wysyła do klienta zawartość pliku config.txt: config (**String**)

S: **send** config => K

- Po uruchomieniu gry Klient zażąda od Serwera wyników graczy

K: „GETSCORES” => S

- W odpowiedzi Serwer wysyła do Klienta zawartość pliku scores.txt: scores (**String**)

S: **send** scores => K

- Po uruchomieniu gry zostaną wysłane żądania pobrania wszystkich planszy poziomów, w postaci „GETMAP i”, gdzie parametr i to numer identyfikacyjny poziomu.

K: „**GETMAP i**” => S

- W odpowiedzi na takie żądanie Serwer odsyła nam zawartość odpowiedniego pliku ze schematem poziomu: lvl (**String**)

S: **send** lvl => K

- Po zakończeniu gry w trybie online i naciśnięciu przycisku Send score na Serwer zostanie wysłane żądanie w postaci:

K: “**PUSH nick score**” =>S

- .Serwer w odpowiedzi wyśle potwierdzenie odebrania wyniku.

S: **send** „OK” => K