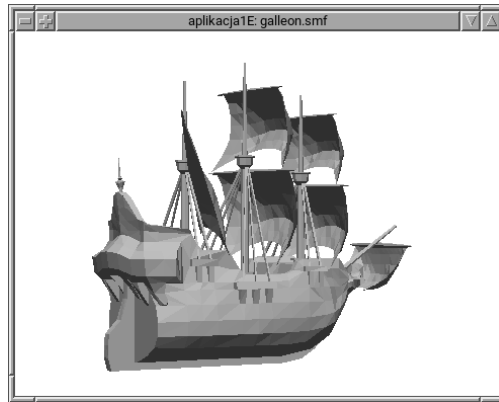


Ostatnie dwie zmiany, których zamieszczanie tu na listingu byłoby przesadą, to zastąpienie w procedurze `DisplayFunc` wywołania procedury `DrawIcosahedron` wywołaniem procedury `DrawSMFObject` i stosowna modyfikacja procedury sprzątania (`Cleanup`).



Rysunek 13.1. Okno aplikacji pierwszej E

13.3. Ćwiczenia

1. Jeśli liczba wierzchołków obiektu nie przekracza 2^{16} , to można oszczędzić połowę miejsca w pamięci GPU zajmowanego przez indeksy tych wierzchołków. W tym celu trzeba przepisać indeksy do tablicy, której elementy są typu `GLushort`, a następnie przesłać do bufora zawartość tej tablicy. Zmień aplikację tak, aby dokonywała tej oszczędności⁴.
2. Przeczytaj opis formatu plików SMF [17] i rozbuduj procedurę czytającą te pliki o wprowadzanie dodatkowych danych: kolorów wierzchołków i wektorów normalnych. Następnie rozszerz aplikację tak, aby te dane były używane podczas wyświetlania obiektów i wypróbuj w działaniu.
3. Utwórz tablicę indeksów wierzchołków, w której kolejne pary liczb określają końce krawędzi trójkątów przeczytanych z pliku SMF, przy czym każda krawędź powinna mieć tylko jedną reprezentację w tej tablicy. Użyj tej tablicy do narysowania krawędzi.

Wskazówka: W każdej parze pierwsza liczba powinna być mniejsza, po posortowaniu par w tablicy można wyeliminować powtarzające się pary.

⁴W „dużych” aplikacjach, wyświetlających sceny zbudowane z wielu skomplikowanych obiektów, takie oszczędności są istotne: nie każda karta graficzna jest wyposażona w 12 GB pamięci. Na przykład moja ma tylko 8 GB.