

# Instrukcja projektowa; projekt 2.

*Podstawy programowania 2017/18 informatyka*

Autor: Paweł Obszarski

Wersja z dnia 21.11.2017

## Projekt Sokoban

### 1. Cel

Napisanie gry planszowej Sokoban. Plansza do gry składa się z kwadratowych pól. Pole może być:

- wolne,
- zajęte przez gracza,
- zajęte przez skrzynię,
- oznaczone jako docelowe,
- ścianą.

Plansza jest zawsze opasana ścianami. Gracz (ludzik) ma możliwość przesuwania się z jednego pola na drugie o ile jest ono wolne. Ponadto gracz może pchać skrzynie (tylko jedną na raz), pod warunkiem, że pole za skrzynią jest wolne. Gracz nie może ciągnąć skrzyń oraz przesuwać ich na boki. Gra polega na przesuwaniu skrzyń w magazynie w taki sposób by wszystkie skrzynie zostały ustawione na miejscach docelowych. Bardziej szczegółowy opis gry można znaleźć w <https://en.wikipedia.org/wiki/Sokoban>.

### 2. Środowisko

Do instrukcji dołączony jest program startowy w którym zaimplementowano:

- obliczanie przyrostu czasu, co pozwala śledzić jego upływ,
- wyświetlanie na ekranie plików graficznych w formacie BMP,
- rysowanie piksela, linii, prostokąta,
- wyświetlanie tekstu.

Program działa w oparciu o bibliotekę SDL2 (2.0.7) – <http://www.libsdl.org/>. Jest ona dołączona do projektu startowego i nie trzeba pobierać jej źródeł.

Kompilacja pod systemem Linux wykonujemy za pomocą komendy (w systemie 32-bitowym):

```
g++ -O2 -I./SDL-2.0.7/include -L. -o main main.cpp -lm -lSDL2 -lpthread -ldl -lrt
```

oraz (w systemie 64-bitowym)

```
g++ -O2 -I./SDL-2.0.7/include -L. -o main main.cpp -lm -lSDL2x64 -lpthread -ldl -lrt
```

W celu pomyślnej kompilacji projektu startowego, w katalogu, w którym znajduje się plik main.cpp powinny znajdować się

- Bitmapy z wymaganymi rysunkami (cs8x8.bmp, eti.bmp). Uwaga na wielkość liter w nazwach plików!
- libSDL2.a (libSDL2x64.a przy kompilacji 64 bitowej)
- Katalog sdl dołączony do projektu.

Do projektu dołączone zostały skrypty, które mogą być użyte do kompilacji (comp w środowisku 32-bitowym oraz comp64 w środowisku 64-bitowym).

Prezentacja programu (zaliczenie tej części projektu) odbywać się będzie w wybranym przez studenta środowisku spośród dwóch poniższych opcji:

- w systemie Linux. Student jest zobowiązany sprawdzić przed przybyciem na zaliczenie czy program poprawnie się kompiluje i uruchamia pod dystrybucją dostępną w laboratorium,
- w systemie Windows, w środowisku MS Visual C++ w wersji zgodnej z tą dostępną w laboratorium.

### 3. Specyfikacja

*Funkcjonalności podstawowe za 5 pkt. (wszystkie muszą być zaimplementowane by dostać jakiekolwiek punkty):*

1. Wyświetlanie planszy w estetyczny i ergonomiczny sposób.
2. Wyświetlanie upływającego czasu podczas rozgrywki. Czas jest zerowany po wczytaniu nowej planszy.
3. Przesuwanie ludzika klawiszami strzałek po wolnych polach bez naruszania ścian, ruch następuje natychmiast – reagowanie na zdarzenia. Ludzik nie może wchodzić w ściany i skrzynie (skrzynie oczywiście można przesunąć).
4. Obsługa klawiszy Esc i n. Gdzie Esc powoduje zamknięcie programu zaś n wczytanie aktualnej planszy od nowa.

*Funkcjonalności zaawansowane za 10 pkt.:*

1. Przesuwanie skrzyń przez popychanie ich. Przesuwanie skrzyni jest możliwe, jeśli pole za nią jest wolne. Nie można przesunąć dwóch skrzyń na raz. 2 pkt.
2. Animowane przejście między polami. 2 pkt.
3. Wczytywanie planszy z pliku. 1 pkt.
4. Menu pozwalające 2 pkt:
  - wczytać gotowe plansze na różnych poziomach. Lista dostępnych plansz powinna być widoczna w menu. Informacja, które pliki są dostępne, nie powinna znajdować się w kodzie programu. Można utworzyć plik konfiguracyjny z dostępnymi planszami.
  - dodanie planszy użytkownika. Gracz podaje nazwę pliku dla własnej planszy.
  - rozegranie wielu gier (jedna po drugiej) podczas jednego uruchomienia programu.
5. Rozpoznanie poprawnego rozwiązania planszy powoduje zakończenie danej rozgrywki i wyświetlenie odpowiedniego komunikatu. 1 pkt.
6. Zapamiętywanie i wyświetlanie z poziomu menu listy najlepszych wyników dla poszczególnych plansz. Lista jest ograniczona wyłącznie rozmiarem pamięci (dynamiczna alokacja pamięci). Archiwizowane wyniki to liczba ruchów i czas rozwiązywania. Gracz ma możliwość dopisania się do rankingu po rozwiązaniu planszy. Ranking powinien być wczytywany i zapisywany w pliku tj. dostępny pomiędzy różnymi uruchomieniami gry. Menu umożliwia wyświetlenie rankingu dla poszczególnych plansz posortowanego po czasie oraz drugiego posortowanego po liczbie ruchów. 2pkt

*Funkcjonalności rozszerzające (sprawdzone tylko jeśli wszystkie powyższe funkcjonalności zostały zrealizowane):*

1. Sterowanie myszką. Kliknięcie dowolnego pola na planszy, które jest osiągalne z aktualnego pola (bez przesuwania skrzyń), powinno spowodować animowane przejście ludzika do tego pola. Jeśli ludzik stoi przy skrzyni kliknięcie na skrzynię powoduje przesunięcie skrzyni (o ile pole za jest wolne) i wejście ludzika na to miejsce.
2. Punktowe oświetlenie planszy/ludzik doświetla planszę latarką. Ludzik, ściany i skrzynie rzucają cienie.

### 4. Uwagi końcowe

1. Zakazane jest korzystanie z biblioteki stl.
2. Kompilacja i uruchomienie programu jest warunkiem przyznania punktów. Student musi sprawdzić czy w sali, w której będzie oddawał jest odpowiednie oprogramowanie.
3. Projekt musi być wykonany samodzielnie przez oddającego studenta. Oddający musi być w stanie wytłumaczyć każdy element kodu oraz nanieść drobne poprawki.
4. Gra powinna być grywalna i wygodna w obsłudze.