

# DOKUMENTACJA PROJEKTU NR 3

Temat - "Samotnik"

Program jest projektem laboratoryjnym z przedmiotu PRI

16.01.2018, Kacper Zając

## ZAŁOŻENIA PROGRAMU

Program jest grą logiczną, w którą jednoczesną rozgrywkę może prowadzić tylko jedna osoba. Plansza do gry składa się z 33 pól, a przy początkowym rozstawieniu na każdym, z wyjątkiem środkowego pola, znajdują się pionki symbolizowane przez program literą „o”. Program objaśnia użytkownikowi cel gry oraz zaznacza go ze sterowaniem podstawowym, a także z dodatkowymi skrótami klawiszowymi. Program zlicza ilość pionków jakie pozostały na planszy, jak i ilość możliwych ruchów do wykonania w danym ustawieniu pionków. Program rozpoznaje zakończenie gry.

Plansza, pionki oraz ruchy są skonstruowane przy użyciu tablic, gdzie główna tablica i układ pionków jest zdefiniowany globalnie. Program przy każdym ruchu użytkownika od nowa wypisuje stan planszy.

Program jest kompatybilny nie tylko z systemem Linux, ale także z Windowsem. Jedynym problemem kompatybilności jest brak wsparcia linuxowej wersji dla zapisów wykonanych na systemie Windows. W drugą stronę wszystko działa bez zarzutu.

## Skróty klawiszowe

Program potrafi:

- zrestartować bieżącą partię,
- zapisać stan gry oraz wczytać go z uprzednio stworzonego pliku. Plik do zapisu można również stworzyć samodzielnie, jednak wymagana jest do tego ścisła kolejność,
- zakończyć działania programu przy pomocy skrótu klawiszowego,
- uruchomić tryb administratora, dzięki któremu można w dowolny sposób ustawiać nowe pionki na planszę lub zabierać te, które się na niej znajdują,
- wyświetlić podpowiedzi dotyczące gry (Rys.1).

```
samotnik
-----

  o o o
  o o o
o o o o o o
o o o<.>o o o
o o o o o o
  o o o
  o o o

Na planszy zostalo 32
Liczba dostepnych ruchow - 4

Nacisnij S, aby wyswietlic pomoc.

samotnik
-----

  o o o
  o o o
o o o o o o
o o o<.>o o o
o o o o o o
  o o o
  o o o

Na planszy zostalo 32
Liczba dostepnych ruchow - 4

Sterowanie:
Po planszy gracz porusza sie przy pomocy strzalek i przycisku Enter.

Przyciski:
R - restart, E - wyjscie z gry, Z - zapisuje stan gry
X - wczytuje stan gry z pliku, A - wlacza tryb administratora
S - zamyka odpowiedzi

Zwyciestwo:
Celem gry jest zostawienie na planszy jak najmniejszej liczby pionkow.
Pionka bije sie przeskakujac go w pionie lub poziomie.
Nie mozna poruszac sie na ukos i bic kilku pionkow jednoczesnie.
```

Rysunek 1

## FUNKCJE

0. *getch()* – funkcja występująca tylko w wersji na system operacyjny Linux. Zastępuje ona jej odpowiednik na system Windows, który zawiera się w bibliotece *conio.h*. Funkcja ta odczytuje kod znaku wciśniętego na klawiaturze.
1. *get\_key()* – zamienia wciśnięty klawisz na jego wartość numeryczną, którą następnie zwraca. Program zwraca inne wartości w zależności od systemu operacyjnego
2. *arrows()* – zaznacza pole, na którym obecnie znajduje się użytkownik.
3. *mark()* – zaznacza pole, z którego użytkownik chce wykonać ruch.
4. *clear\_arrows()* – odznacza wyżej wymienione pola.
5. *count\_remaining()* – liczy ile ruchów można wykonać przy danym ustawieniu pionków.
6. *restart()* – przywraca początkowy stan planszy.
7. *save\_file()* – zapisuje stan gry w wybranym przez użytkownika adresie. Dodatkowo zapisuje obecną pozycję gracza na planszy oraz ilość pozostałych pionków. Jeśli plik już istnieje pyta użytkownika, czy chce ten plik nadpisać.
8. *open\_file()* – wczytuje wcześniej zapisany stan gry. Rozpoznaje czy plik istnieje.
9. *show\_board()* – jedna z najważniejszych funkcji programu. Czyści terminal, wypisuje całą planszę i wszystkie inne informacje.
10. *relation()* – zwraca 1 jeśli możliwy jest ruch w dowolnym kierunku z obecnej pozycji gracza.
11. *jump()* – funkcja ta odpowiada za zbijanie pionków.
12. *end()* – sprawdza czy, po skończonej partii, użytkownik chce grać dalej, czy woli zakończyć rozgrywkę.

13. *move()* – największa funkcja programu. Odpowiada z ruch po planszy. Jest skonstruowana przy pomocy instrukcji wyboru switch. Określa co należy wykonać po wciśnięciu każdego z przycisków. Określa także z jakich pozycji wolno wykonać bicie. Jest zapętłona sama w sobie, a wyjście z niej oznacza zakończenie gry (schemat 1).

## ZMIENNE GLOBALNE

- `char board[7][16]` – tablica planszy do gry.
- `int current_x` – obecna pozycja na osi x,
- `int current_y` – obecna pozycja na osi y,
- `int remaining_peg` – ilość pozostałych na planszy pionków,
- `int remaining_moves` – ilość możliwych do wykonania ruchów,
- `_Bool error` – prawda jeśli gracz próbował wykonać ruch z miejsca niedozwolonego,
- `_Bool admin_on` – prawda jeśli włączony jest tryb administratora,
- `_Bool howtoplay` – prawda jeśli włączona jest instrukcja.
- `FILE *file` – wskaźnik na otwierany/zapisywany plik.

## TESTOWANIE

Program testowałem zarówno podczas wielokrotnych rozgrywek, jak i sprawdzając skrajne przypadki przy użyciu trybu administracyjnego. Między innymi:

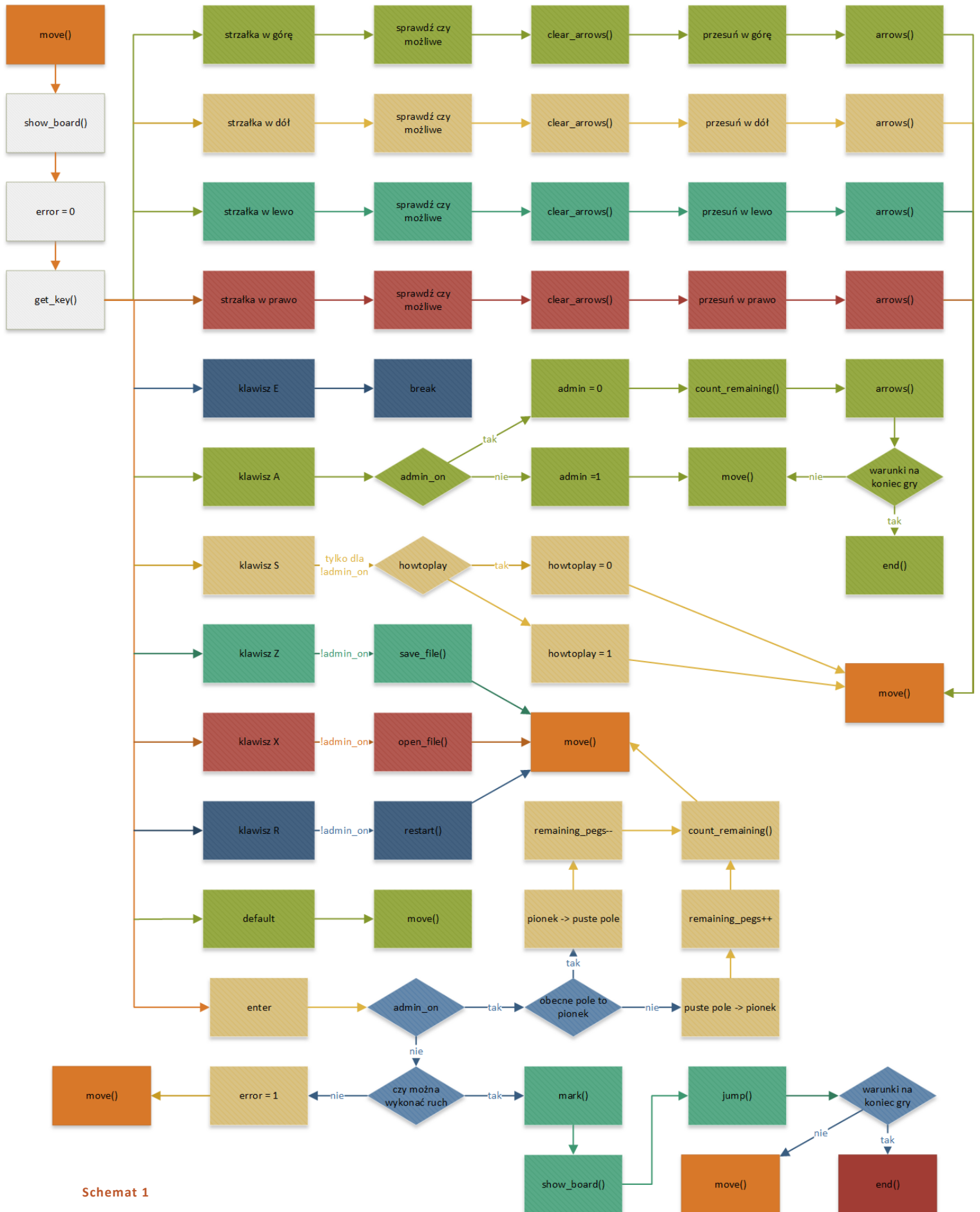
- usunięcie wszystkich pionków,
- usunięcie wszystkich z wyjątkiem jednego,
- usunięcie wszystkich wolnych pól.

Z moich obserwacji wynika, że niemożliwe jest spowodowanie błędu (za wyjątkiem wcześniej wspomnianej próby wczytywania zapisu gry z Windowsa na Linuxie). W każdym z powyższych przypadków program poinformuje użytkownika, że gra skończona i zaproponuję kolejną rozgrywkę (jednokrotnie ogłaszając użytkownika zwycięzcą).

Podczas sprawdzania błędów przy zapisie lub wczytywaniu zapisanego stanu gry możliwe błędy to:

- błędny adres,
- używany adres.

Program ma przygotowane komunikaty na oba te przypadki.



Schemat 1