

## Zmienne i struktury danych:

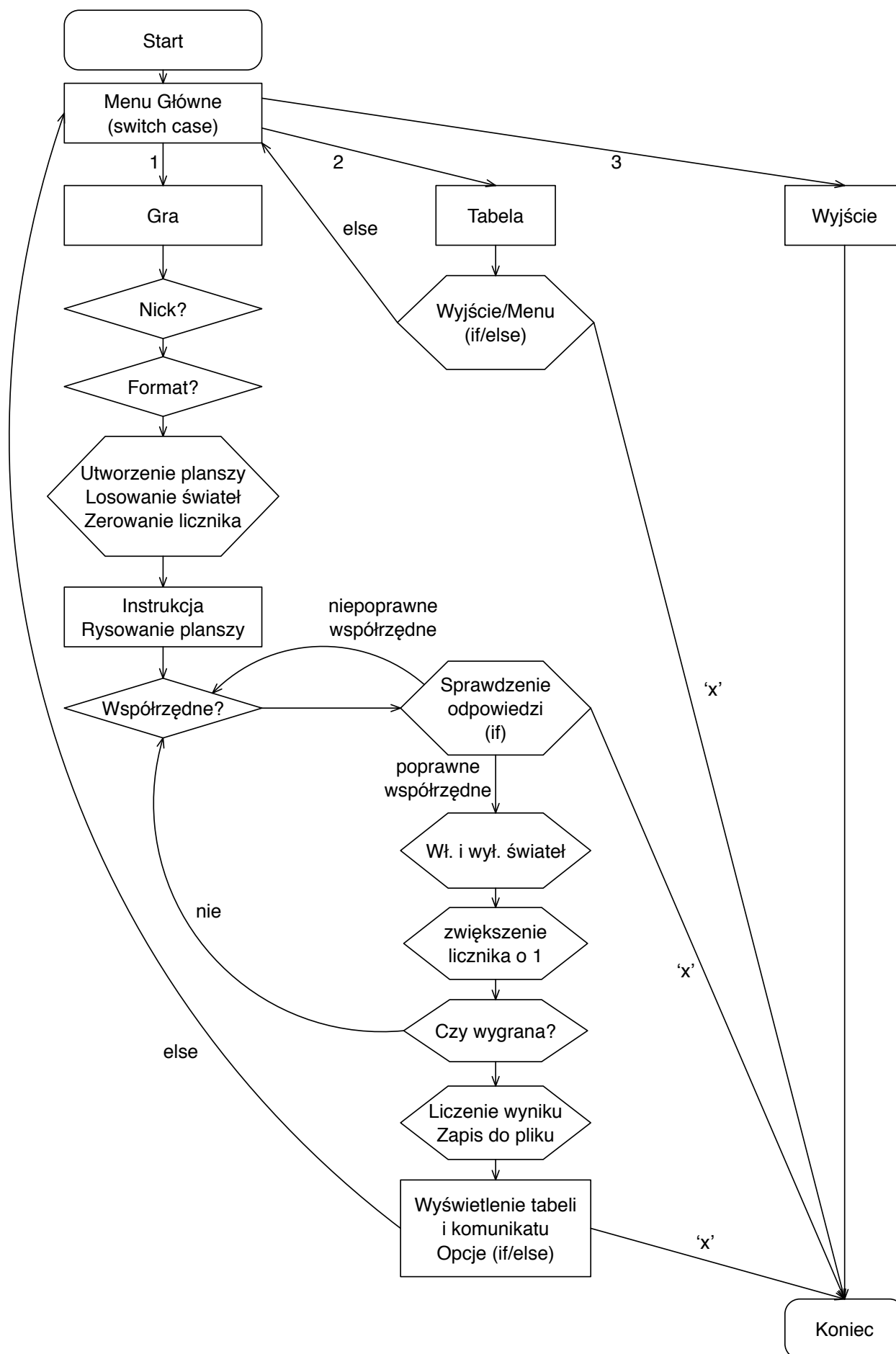
## Zmienne

Nazwa zmiennej	Typ	Funkcja	Opis
<b>N</b>	int	liczba wierszy	Możliwa regulacja rozmiaru planszy
<b>M</b>	int	liczba kolumn	
<b>naglowek</b>	char	regulacja zakresu nazw kolumn	„Podaj wiersz/kolumnę [A-*zmienna*]”
<b>naglowek2</b>	int	regulacja zakresu nazw wierszy	
<b>i</b>	int	pętla for	w tym pętli w pętli
<b>j</b>	int	pętla for	
<b>k</b>	int	pętla for	
<b>l</b>	int	pętla for	
<b>menu</b>	char	wybór opcji w głównym menu	-
<b>powrot</b>	char	wybór opcji powrót do menu	
<b>wiersz</b>	int	indeks wiersza	wpisywane przez użytkownika. Plansza ma nagłówki typu „C4”
<b>kolumna</b>	char	indeks kolumny w formie litery	
<b>nrkolumna</b>	int	indeks kolumny w formie cyfry	po zamianie z litery na cyfrę
<b>zgaszone</b>	int	liczba zgaszonych żarówek	-
<b>zapalone</b>	int	liczba zapalonych żarówek	
<b>wynik</b>	int	wynik gracza	
<b>warunek</b>	int	jeśli spełniony, kończy grę	wszystkie żarówki muszą zostać zgaszone
<b>licznik</b>	int	liczy ilość ruchów	-
<b>rozmiar</b>	int	$(N-1) * (M-1)$	potrzebne do własnej punktacji
<b>gotowe</b>	int	warunek sprawdzenia wyników równych	potrzebne w zapisie do pliku i sortowaniu

## Struktury danych

Nazwa	Typ	Funkcja	Komentarz
<b>P</b>	stała	zdefiniowana na początku stała	stała globalna
<b>name[P]</b>	tabela 1D char	tabela na nick	-
<b>gracz[P][P]</b>	tabela 2D char	tabela na graczy	podczas zapisu do pliku
<b>wynikgracza[P]</b>	tabela 1D char	tabela na wyniki graczy	
<b>„leaderboard.txt”</b>	PLIK	plik do zapisu wyników	-

## Schemat algorytmu:



## Modyfikacje:

1. Możliwość wyboru rozmiaru planszy - na początku gry użytkownik może wybrać dowolny rozmiar planszy w zakresie [wiersz] = <1;10>, [kolumna] = <1;12>.
2. Tablica wyników - po grze i uzyskaniu danego wyniku, program sprawdza, czy wynik jest lepszy bądź równy któremuś z wyników znajdujących się w tabeli rekordów (w pliku). Jeśli tak, ustawia nowy wynik w odpowiednim miejscu i przesuwa pozostałe.

## Opis działania programu:

Celem zadania było napisanie w pełni funkcjonalnej gry logicznej, polegającej na zapalaniu i gaszeniu pól na planszy 10x12 (w przypadku mojej modyfikacji, program pyta również użytkownika o preferowany rozmiar planszy - zastrzega jednak, że im mniejsza będzie plansza, tym mniej punktów uzyska), wykorzystującej terminal do wyświetlania planszy i interfejsu użytkownika. Gra miała polegać na gaszeniu i zapalaniu odpowiednich żarówek. Na początku gry położenie wszystkich żarówek jest generowane losowo, a licznik ruchów jest wyzerowany. Gracz powinien wprowadzić współrzędne punktu, w którym chce zgasić/zapalić żarówkę. Po tym, podane przez gracza pole, a także sąsiadujące z jego krawędziami 4 inne pola, zmieniają swoją wartość na przeciwną (są zapalane lub gaszone w zależności od poprzedniego stanu). Licznik ruchów zwiększa się o jeden, program sprawdza warunek wygranej. Celem jest doprowadzenie do stanu, kiedy wszystkie żarówki na planszy są zgaszone. Jeżeli nie są - gra trwa dalej. Wynik gracza zależy od ilości wykonanych ruchów. Program ten musi być odporny na błędnie wprowadzone dane, a użytkownik ma mieć możliwość wyjścia z gry w dowolnym momencie. Moja druga modyfikacja polega na tym, że po zakończeniu gry program sprawdza wynik danego gracza i porównuje go z listą rekordów z pliku „leaderboards.txt”. Jeżeli wynik jest słabszy, ignoruje go, a jeśli jest lepszy, wpisuje go na odpowiednie miejsce tablicy wyników. Program jest kończony wtedy, kiedy użytkownik go zamknie według instrukcji na ekranie.

## Opis czynności w punktach, [główne funkcje]:

### ● Start programu [pętla for( ; ; )]

### ● - Menu Główne: [switch( ) case]

#### • 1 - Nowa Gra: [case 1] [do( ), while(warunek == 0/1)]

- Tworzenie pliku do zapisu, jeśli nie istnieje [fopen(„\*.txt“, „ab+“), [fclose]
- Program prosi o podanie nicku [scanf]
- Program prosi o podanie rozmiaru planszy [scanf]
  - *(W tle tworzy się plansza o podanym rozmiarze z odpowiednimi nagłówkami, losuje się położenie zapalonych żarówek, zeruje licznik ruchów)* [srand(time(0)) + rand( )%2]
- Wyświetla się instrukcja gry i odpowiednia plansza, a także licznik ruchów i informacja „jak wyjść z gry“
- Program prosi o podanie współrzędnej kolumny [scanf]
- Program prosi o podanie współrzędnej wiersza [scanf]
  - *(W tle podane dane zamieniają się na odpowiednie współrzędne tabeli)* [switch( ) case]
- Zwiększa się licznik ruchów
- Żarówka w podanej lokalizacji (jak i odpowiednie 4 inne) gaśnie/zapala się [if, else]
  - *(W tle program sprawdza warunek wygranej)* [warunek = 0 lub warunek = 1]
- Jeśli warunek jest spełniony: [(warunek == 1)]
  - *(W tle program liczy wynik w zależności od rozmiaru planszy i liczby ruchów)* [if, else if], [switch( ) case]
  - Jeśli warunek nie jest spełniony program ponownie prosi o podanie współrzędnych i gra toczy się dalej... [(warunek == 0)]
- Wyświetla się plansza, na której wszystkie żarówki są już zgaszone
- Wyświetla się informacja o wygranej i uzyskanym wyniku
  - *(W tle program wczytuje z pliku listę rekordów, sprawdza czy i gdzie umiejscowić wynik danego gracza, zapis do pliku)* [fopen], [fscanf], [fprintf] [if, else if]
- Wyświetla się informacja o powrocie do Menu Głównego lub zakończeniu programu

#### • 2 - Tabela wyników [case 2]

- Wyświetla się aktualna tablica rekordów wczytywana z pliku „leaderboards.txt“

#### • 3 - wyjście z programu [case 3]

### ● Koniec programu