

Instrukcja użytkownika programu Sokoban

- Cel gry:**

Sokoban to gra logiczna, w której celem gracza jest ustawienie skrzyń w specjalnie przeznaczonych do tego punktach. W trakcie rozgrywki gracz musi uważać na swojego przeciwnika (duszka), którego zadaniem jest utrudnienie rozgrywki. W przypadku spotkania się duszka z bohaterem, nasza postać traci punkt życia. W trakcie gry użytkownik dysponuje trzema żywami. Utrata ich wszystkich oznacza przegraną i powrót do menu startowego.

- Możliwość personalizacji gry:**

Gra Sokoban posiada liczne możliwości personalizacji. Gracz może wybrać tryb rozgrywki- lokalny albo sieciowy, podać nazwę użytkownika czy wybrać awatar postaci. Dodatkowo gra, względem oryginału, posiada odświeżoną szatę graficzną, dzięki czemu jest bardziej przyjazna dla użytkownika. Względem oryginału została także wprowadzona zmiana dotyczącą duszka, który próbuje utrudnić grę.

- Uruchomienie gry:**

Gra może działać w dwóch trybach rozgrywki- lokalnym albo sieciowym.

Uruchomienie gry:	
Tryb lokalny	Tryb sieciowy
1. Uruchomić klienta poprzez wcześniej przygotowane pliki bat-kompilacja oraz uruchomienie. Po uruchomieniu na ekranie pojawi się menu gracza.	1. Skompilować i uruchomić serwer (do tych czynności mogą posłużyć specjalnie przygotowane programy zawarte w podkatalogu Serwer- kompilacja oraz uruchomienie). 2. Z uwagi na połączenie serwera i klienta za pomocą numeru portu oraz hosta serwer na samym początku automatycznie generuje plik, gdzie są zawarte wszystkie niezbędne informacje do zaczącia gry poprzez serwer. Wygenerowany plik (DaneWyjscioveSerwera.txt, który znajduje się w tym samym miejscu co programy do kompilacji i uruchomienia serwera) należy przenieść do folderu Klient->src->com->sokoban->config w celu możliwości odczytania go przez klienta. 3. Uruchomić klienta i wybrać interesującą nas opcję (do tej czynności również można użyć wcześniej przygotowane pliki kompilacja oraz uruchomienie).

Tab. 1- uruchomienie gry

Uwaga! Domyślnie serwer pracuje na numerze portu 5023. Jeśli podczas uruchomienia okazałoby się, że port ten jest zajęty, to jest możliwość jego zmiany w pliku Serwer->Pliki->Port.txt. W takim przypadku należy powtórzyć czynności opisane w podpunkcie uruchomienie gry w trybie sieciowym (Tab. 1).

- **Wymagania systemowe do uruchomienia gry:**

Do uruchomienia gry jest niezbędne posiadanie środowiska Javy (JRE) lub ewentualnie wszystkie potrzebne pliki zawarte w pakiecie JDK, by móc samemu skompilować i wykonać program. Aplikacja JDK wymaga wersji 11.

- **Zasady:**

Gra składa się z określonej liczby poziomów. Każdy poziom jest predefiniowany w plikach konfiguracyjnych. Po pomyślnym rozłożeniu wszystkich skrzyń gra przechodzi do kolejnego poziomu, a po ukończeniu wszystkich następuje koniec gry. W przypadku nieudanej próby gracz może zrestartować grę wciskając klawisz z klawiatury „R”. Gracz musi uciekać przed swoim przeciwnikiem (duszką). Spotkanie się z duszką powoduje utratę jednego z trzech, początkowo dostępnych żyć. W przypadku utraty wszystkich żyć gra się kończy przegraną.

- **Zasady punktacji:**

Na ostateczną punktację gracza składa się kilka czynników, których średnia ważona jest liczbą punktów uzyskanych przez gracza. Czynniki wliczające się w wynik gracza:

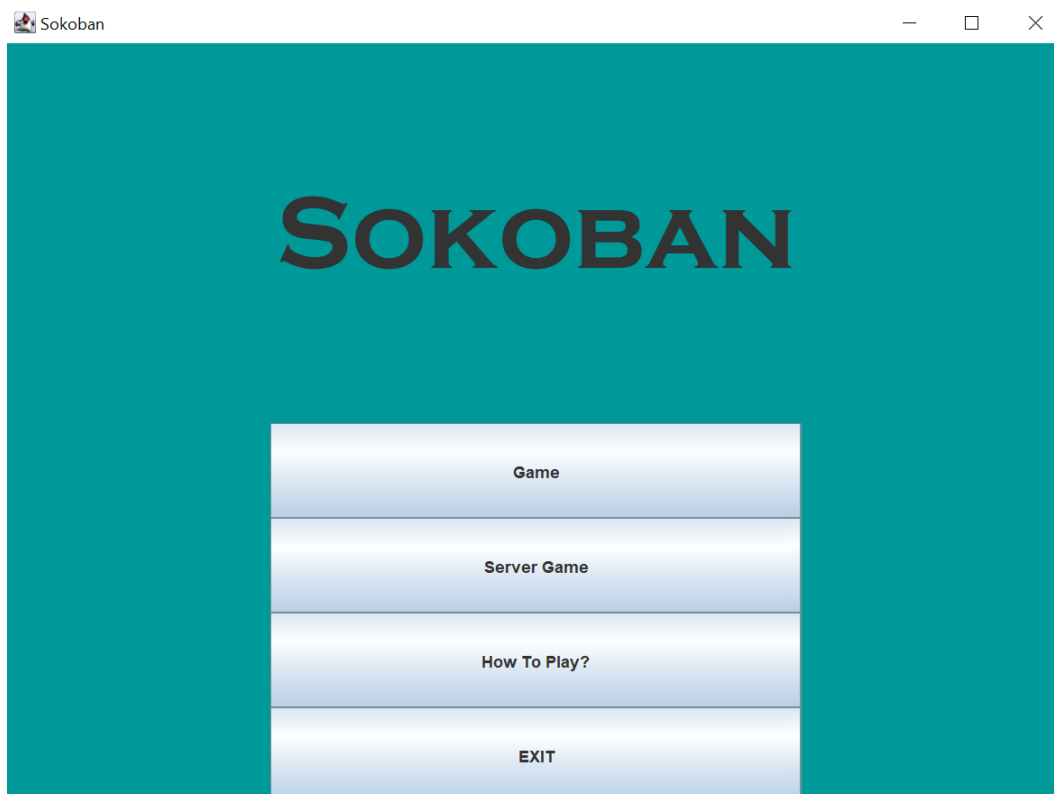
1. Ilość użytych restartów (waga 3),
2. Ilość kroków (waga 1),
3. Ilość straconych żyć (waga 20),
4. Ilość nieukończonych poziomów (waga 1).

Zadaniem gracza jest minimalizacja liczby wyżej wymienionych parametrów. Wraz ze zmniejszeniem parametrów gracz otrzymuje wyższy wynik końcowy.

- **Sterowanie:**

Gracz steruje swoją postacią za pomocą strzałek na klawiaturze. Przeciwnicy duszka poruszają się w sposób losowy, dlatego niekiedy trudno uniknąć z nimi kontaktu.

- **Opis interfejsu gracza:**



Ryc. 1 Ekran główny aplikacji

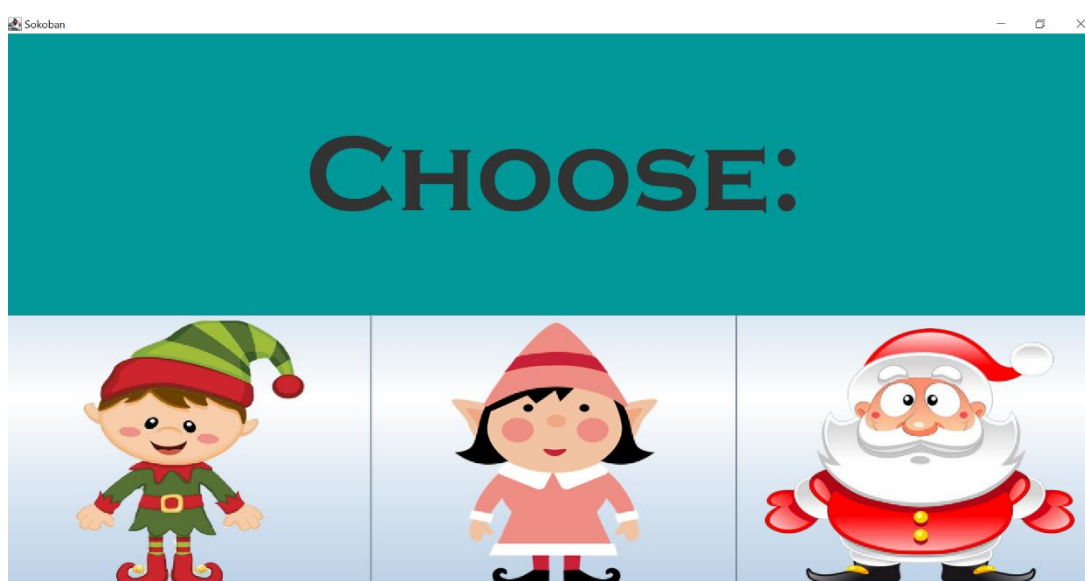
Po uruchomieniu gracz ma możliwość wyboru jednego z dwóch trybów gry- sieciowej lub lokalnej. Gracz może również zobaczyć informację o grze w sekcji „How to Play?”. W przypadku wyboru opcji „Server Game” użytkownik loguje się na (dane te nie są stałe dlatego osoba zarządzająca serwerem musi podać je użytkownikom). Wybór opcji „Game” to tryb lokalny. Po wybraniu trybu online aplikacja będzie wczytywać poziomy, listę najlepszych wyników, listę konfiguracyjną oraz elementy graficzne z serwera. W przypadku trybu offline powyższe dane zostaną wczytane z plików lokalnych.

Uwaga! Gra online jest możliwa wyłącznie po wcześniejszej kompilacji i uruchomieniu serwera.



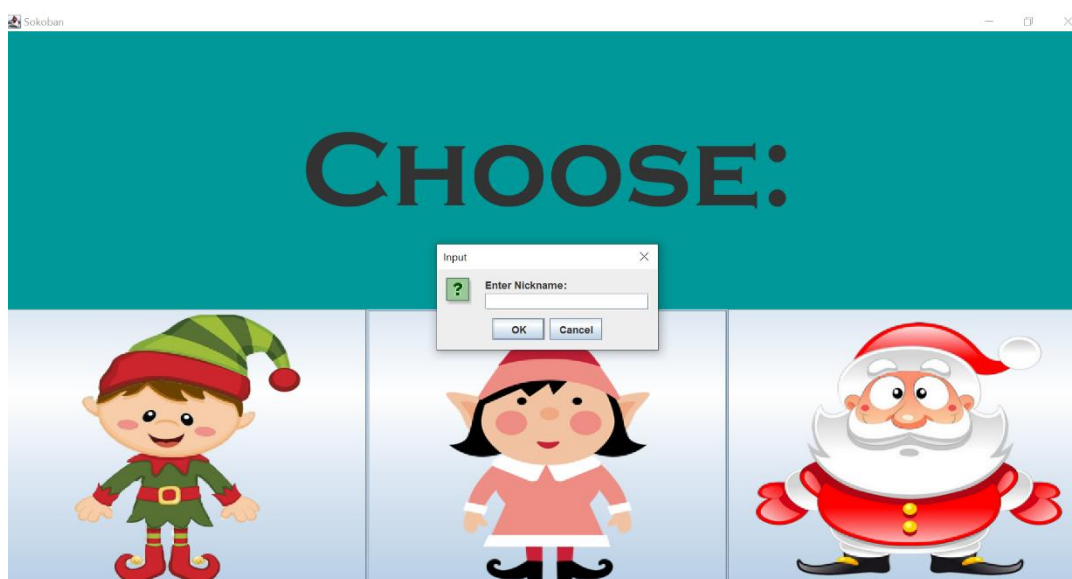
Ryc. 2 Podmenu aplikacji

Niezależnie od wybranego trybu aplikacja przejdzie do menu gry w którym gracz może rozpocząć rozgrywkę, zobaczyć listę najlepszych wyników albo wyjść do menu głównego.



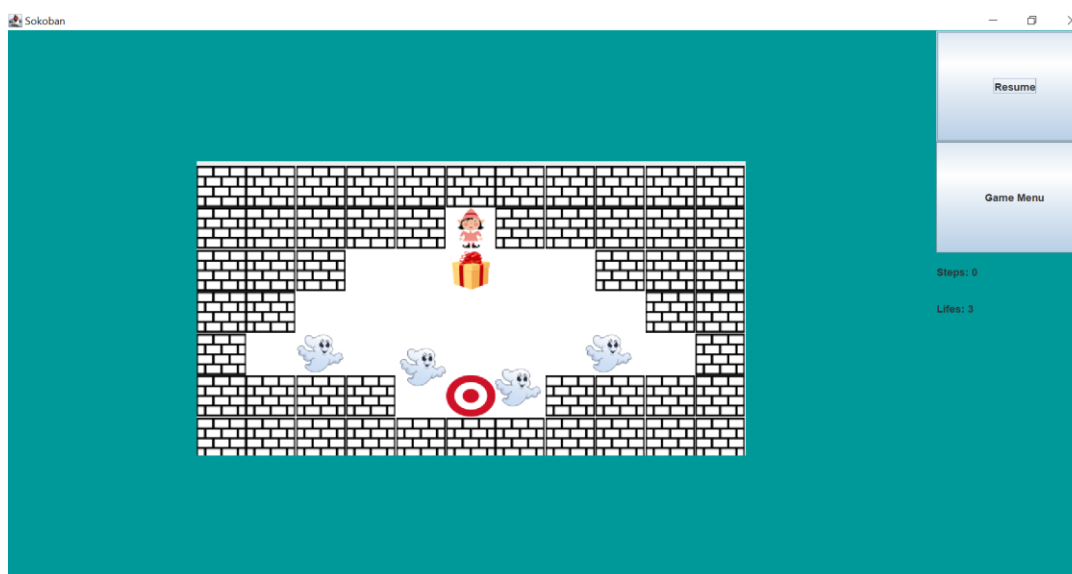
Ryc. 3 Wybór awataru

Po wybraniu opcji „Start” gracz ma możliwość dokonania personalizacji swojego bohatera poprzez wybranie awataru postaci. Wybór jest dokonywany za pomocą myszki.



Ryc. 4 Wprowadzenie nazwy gracza

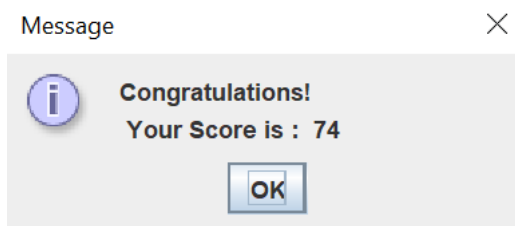
Po wybraniu bohatera gracz podaje swoje imię które potencjalnie może zostać zapisane na liście najlepszych wyników i naciska przycisk „Ok” w celu rozpoczęcia rozgrywki.



Ryc. 5 Plansza rozgrywki

Ekran rozgrywki składa się z planszy wraz z bohaterem, duszkiem, prezentami (skrzynkami) oraz docelowym punktem dla tych skrzynek i bocznego menu, w którym wyświetlane są informacje oraz przyciski. Dwa górne przyciski znajdujące się w prawej części ekranu powodują zapauzowanie gry oraz możliwość przejścia do menu gry. Poniżej znajdują się informacje dotyczące liczby wykonanych kroków w trakcie rozgrywki oraz ilości pozostałych żyć.

Po wykonaniu danego poziomu zostanie załadowana kolejna mapa, a parametr kroków i pozostałej ilości żyć zostanie zapamiętany i przekazany do następnego poziomu.



Ryc. 6 Wynik

Po ukończeniu gry zostaje wyświetlony wynik końcowy. Naciśnięcie przycisku „Ok” powoduje powrót do odpowiedniego podmenu.

Rank	Name	Score
1	Krzyś	200
2	Anna	180
3	Kuba	160
4	Jacek	155
5	Basia	140
6	Kacper	130
7	Dariusz	120
8	Piotrek	100
9	Konrad	80
10	Adrian	75
11	Jamusz	55
12	Halina	50
13	Marian	30
14	Wioła	20
15	Krzyszian	10

Ryc. 7 Lista najlepszych graczy.

Istnieje również opcja wyświetlenia listy najlepszych wyników w ramach danego trybu gry. Klikając przycisk „Back” można wrócić do podmenu „Game” lub „Server Game”.

- **Dodatkowe informacje:**

Projekt składa się z dwóch osobnych aplikacji – serwera i klienta. Serwer jest bezobsługową konsolową aplikacją a Klient jest okienkową aplikacją w której odbywa się rozgrywka. W folderach obu aplikacji znajdują się pliki konfiguracyjne w formacie .txt, pliki tekstowe w formacie .png oraz pliki wsadowe systemu Windows służące do kompilacji, uruchomienia programu, i generowania dokumentacji.

1. Pliki konfiguracyjne:

Aplikacja jest w całości sparametryzowana przez pliki konfiguracyjne. Można dowolnie zmieniać scenariusz gry i tworzyć nowe poziomy poprzez modyfikacje wyżej wymienionych plików.

```
3
800
600
0
3
0
2
15|
```

Ryc. 8 Pliki konfiguracyjne.

Opis liczb od góry: ilości żyć (3), wymiarów okien (800,600), ilość kroków na początku gry (0), ilość map (3), pierwsza mapa (0), ostatnia wczytana mapa (2), zmiana szybkości (15).

```
#####
#####@#####
###^ $ ^###
##^      ^##
#         #
##### . #####
#####
```

Ryc. 9 Przykładowa mapa

Z przedstawionego schematu tworzona jest mapa (# to ściana, @ to punkt startowy postaci, \$ to bloki do poruszenia, . to punkt docelowy, ^ to przeciwnicy).



Boy



Girl



Santa



Box1



Box2



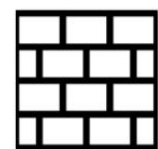
Evil



Magnet



Target



Wall

Ryc. 10 Tekstury i awatary

Tekstury i awatary są przechowywane w odpowiednich dla siebie folderach, w postaci plików png. Użytkownik może dowolnie spersonalizować grę poprzez zmianę tych atrybutów.

```
Krzyś 200  
Anna 180  
Kuba 160  
Jacek 155  
Basia 140  
Kacper 130  
Dariusz 120  
Piotrek 100  
Konrad 80  
Adrian 75  
Janusz 55  
Halina 50  
Marian 30  
Wiola 20  
Krystian 10
```

Ryc. 11 Ranking

Ranking jest przechowywany w odpowiednim dla siebie folderze, w postaci plików txt. Użytkownik może zostać wpisany na listę najlepszych graczy jeśli uzyska wynik lepszy od najgorszego znajdującego się na liście. Każda linia tekstu składa się z nicku gracza oraz liczby punktów jakie otrzymał w trakcie gry.

2. Serwer:

Serwer jest bezobsługową konsolową aplikacją i jedynie wyświetla adres IP maszyny na której jest uruchomiony oraz numer portu (numer portu Jest przechowywany w pliku konfiguracyjnym), na którym działa. Gdyby się zdarzyło, że numer portu jest zajęty aplikacja rzuci wyjątek i nie będzie dalej działać. W takim wypadku należy zmienić numer portu. Dodatkowo, na bieżąco wyświetlane są żądania klienta oraz odpowiedzi serwera. Serwer posiada własne pliki konfiguracyjne w tym samym formacie, co pliki klienta. Jedyną różnicą jest dodatkowy parametr, będący numerem portu. Poniżej zamieszczono przykładowy zrzut ekranu z działania serwera.


```
Waiting for connection!
From client: Load_config
Server respond: 3/800/690/0/4/0/3/
From client: Load_levels
Server respond: /#####Wq$.#/#####
From client: Load_levels
Server respond: /#####/#. .#/# * # #/# $ #/# $q$ #/# $ #/# * # #/#. .#/#####
From client: Load_levels
Server respond: /#####/#. .#/#.$$$.#/##$@$##/#. $$$.#/#. .#/#####
From client: Load_levels
Server respond: /#####/#. .#/# $q$ #/# ## #/# $ $ #/#. .#/#####
From client: Load_textures
Pliki/textures0.png
From client: Load_textures
Pliki/textures1.png
Sorry! We could not do command!
java.io.EOFException
From client: Load_textures
Pliki/textures2.png
From client: Load_textures
Pliki/textures3.png
From client: Load_textures
Pliki/textures4.png
From client: Load_textures
Pliki/textures5.png
From client: Load_avatars
Pliki/avatars\Boy.png
From client: Load_avatars
Pliki/avatars\Girl.png
From client: Load_avatars
Pliki/avatars\Santa.png
From client: Load_ranking
Server respond: Mariusz 130/Dariusz 120/Piotrek 100/Pola 98/Konrad 80/Adrian 75/Janusz 55/Halina 50/Marian 30/Wiola 20/
```

Ryc. 12 Przykładowa komunikacja z serwerem