Oleksii Perets

104156

Game advisor

# Oświadczenie

Świadomy odpowiedzialności prawnej oświadczam, że niniejszy projekt został napisany przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego lub zaliczenia przedmiotu z żadnej uczelni ani szkoły.

Spis treści

[Oświadczenie 1](#_Toc535335179)

[Manual 2](#_Toc535335180)

[Zadanie, które program ma realizować 2](#_Toc535335181)

[Lista opcji do wyboru z krótkim opisem każdej z nich 2](#_Toc535335182)

[Składnia danych wejściowych i wyjściowych 5](#_Toc535335183)

[Struktura plików, typy danych przechowywane w poszczególnych polach pliku 5](#_Toc535335184)

[Opis kodu 6](#_Toc535335185)

[Lista plików z kodem źródłowym wchodzących w skład programu 6](#_Toc535335186)

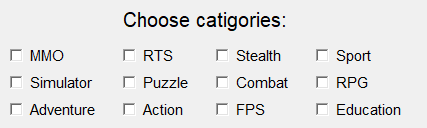
[Diagram klas 6](#_Toc535335187)

# Manual

## Zadanie, które program ma realizować

Użytkownik podaje programowi gatunek/gatunki gier (RPG, stealth, sport itd…). Program zwraca najlepsze gry pasujące pod podane gatunki. Użytkownik może dodawać i usuwać gry z bazy wiedzy, oraz zmieniać jak gry są wpisane w bazie wiedzy.

## Lista opcji do wyboru z krótkim opisem każdej z nich



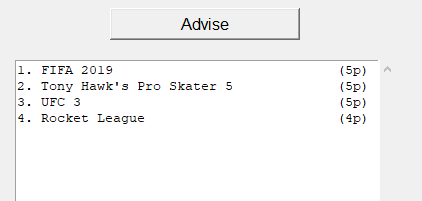
Użytkownik zaznacza dowolną ilość gatunków w których jest zaciekawiony.

Następnie można wybrać jedną z dwóch opcji: bramka „AND” lub bramka „OR”. Domyślnie jest zaznaczona bramka „AND”.

Bramka „AND”: wyszukuje gry które mają nie minimalną ocenę z KAŻDEGO z wybranych przez użytkownika gatunków.

Bramka „OR”: wyszukuje gry które mają nie minimalną ocenę z CO NAJMNIEJ JEDNEGO z wybranych przez użytkownika gatunków.

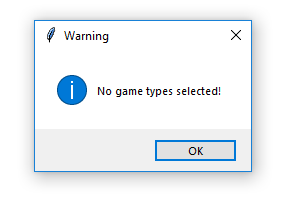
Bramki mają znaczenie tylko przy zaznaczeniu kilku gatunków (>1), dla jednego gatunku program zwróci dokładnie taką samą odpowiedź.

Po wciśnięciu przycisku „Advise”, program zwróci swoją odpowiedź w specjalnym polu odpowiedzi, gdzie ją można zaznaczyć i skopiować.

Odpowiedź skała się z posortowanej po punktach i ponumerowanej listy:

1. <Nazwa gry> (<punkty>p)

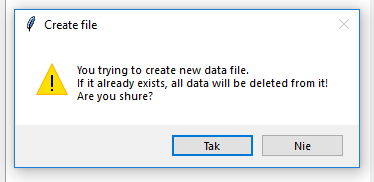
Punkty służą tylko elementem porównania, jaka gra pasuje zaznaczonym gatunkom bardziej, a która mniej, i o ile.



Jeśli klikając „Advise” użytkownik niema zaznaczonego żadnego gatunku, to pojawi się powiadomienie z odpowiednim komunikatem.

W lewym górnym rogu okna programu znajdują się rozwijane menu „File” i „Data”.

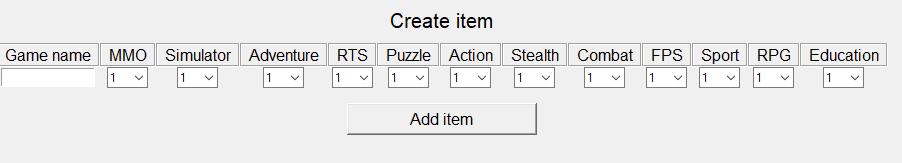
File:

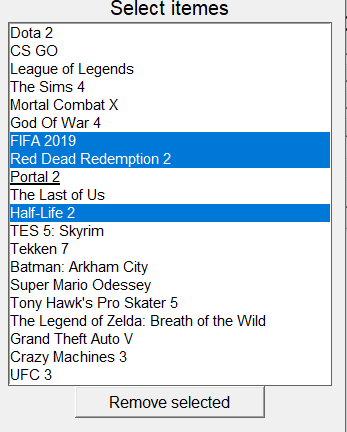
„Create file” – tworzy .csv plik, przygotowany do wypełnienia danymi. Przez tworzeniem pliku, program pyta o potwierdzenie i uprzedza o usunięciu dotychczasowych danych w razie zgody.

„Data file changed” – odświeża dane w programie, danymi z pliku.

Data:

„Add game” – otwiera okno z formularzem, do tworzenia wpisu do pliku z danymi



„Remove games” – otwiera okno z listą wszystkich dostępnych gier z pliku z danymi. Zaznaczając jedną albo kilka gier i następnie klikając „Remove selected”, zaznaczone gry zostaną usunięte.

„Edit games” – otwiera podobne okno z listą gier, z taką różnicą że można zaznaczyć tylko jedną gre, a jedyny przycisk ma nazwe „Edit selected”. Klikając ten przycisk otwiera się kolejne okno, z formularzem jak przy dodaniu nowej gry, tylko już wypełnionym danymi zaznaczonej gry. Dane można zmienić i zapisać zmiany.

# Składnia danych wejściowych i wyjściowych

## Struktura plików, typy danych przechowywane w poszczególnych polach pliku

**Plik: log.txt** <dane wyjściowe>

Aktualna data i czas

„status”

**Przykłady statusów:**

Start programu…

Uzytkownik rozpoczyna nowa gre.

Gracze ustawiaja piony.

….

**Przykład:**

Mon May 29 00:24:18 2017

Start programu...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Plik: Save.txt** <dane wyjściowe \ wejściowe>

Dla jednogo gracza (wpisywane są trzy po kolej):

0\1 (1 – komputer \ 0 - czlowiek) , punkty , ilość pionków

0\1\2 (stopniowo opisana ilość kątów), 0\1\2 (stopniowo opisana wysokość pionka), x , y , orientacja

<opisuje się każdy pionek>

…

**Przykład:**

1,10,3 <gra komputer, ma 10 punktów, ma trzy pionki>

2,2,4,0,1 <sześciokąt, najwyższy, lokalizacja (4,0), orientacja 1>

1,1,1,0,0 <kwadrat, średniej wysokości, lokalizacja (1,0), orientacja 0>

0,0,4,0,0 <trójkąt, najniższy, lokalizacja (4,0), orientacja 0>

# Opis kodu

## Lista plików z kodem źródłowym wchodzących w skład programu

Negłówkowe: Żriódłowe:

color.h color.cpp

figure.h figure.cpp

fildes.h fildes.cpp

gameplay.h gameplay.cpp

global\_vars.h main.cpp

player.h player.cpp

save.h save.cpp

set\_the\_beginning.h set\_the\_beginning.cpp

wypisz\_pola.h wypisz\_pola.cpp

## Diagram klas