MATCH-3

Szymon Spaczyński, Politechnika Warszawska

1. OPIS

Match-3 to gra polegająca na ułożeniu trzech lub więcej kwadratów tego samego koloru w jednej linii (poziomej lub pionowej) poprzez zamianę miejscami dwóch sąsiadujących ze sobą kwadratów. Za każde takie ułożenie gracz otrzymuje punkty zgodnie ze wzorem:

$$l.punktów = 10 * l.ułożonch\ kwadratów * nr\ bieżącego\ poziomu$$

W miarę zbijania kolejnych kwadratów gracz awansuje na kolejne poziomy, a każdy poziom jest trudniejszy od poprzedniego (jest więcej różnych kolorów na planszy). Gra kończy się w momencie, gdy gracz nie ma już żadnego ruchu, który umożliwiałby mu ułożenie co najmniej trzech kwadratów w linii. W tym momencie gracz podaje swoje imię, które – wraz z jego wynikiem – jest zapisywane do pliku result.yml.

2. INSTRUKCJA I WYMAGANIA

Do uruchomienia gry niezbędne będą:

- Interpreter języka Python (zalecana wersja 3.9 lub nowsza)
- Biblioteka Pyside2 (można ją zainstalować wpisując w terminalu komendę *pip install PySide2*)
- Moduł PyYAML (komenda *pip install pyyaml*)

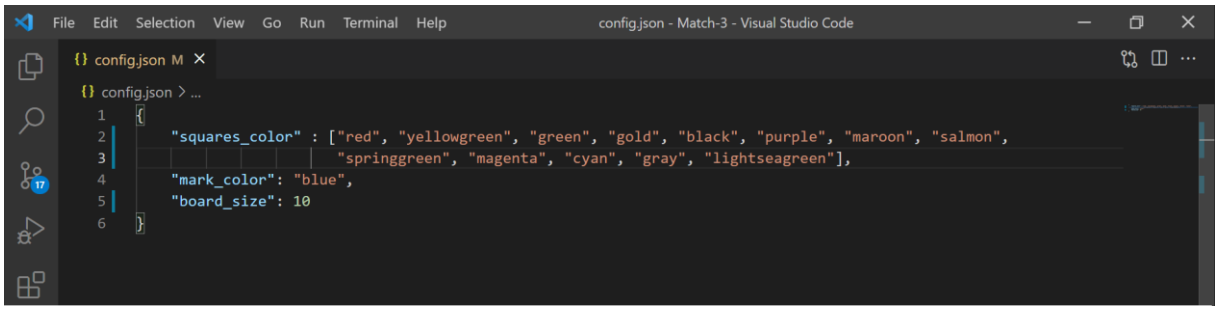
Przed uruchomieniem gry należy upewnić się że w katalogu, którym znajduje się gra, jest umieszczony plik o nazwie *config.json*, zawierający dane konfiguracyjne naszej gry:

- tablicę przypisaną do klucza *squares_color* z kolorami kwadratów, które pojawią się na planszy. Musi być **co najmniej 13 różnych kolorów** (jeśli będzie mniej, próba uruchomienia gry zakończy się błędem). Nazwy kolorów muszą być zgodne ze standardami kolorów w CSS – można użyć zapisu heksadecymalnego w standardzie RGB (#RRGGBB), lub użyć jednej ze 140 obsługiwanych nazw: https://www.w3schools.com/colors/colors_hex.asp

  Zalecane jest użycie kolorów różniących się w sposób widoczny odcieniami, tak aby nie utrudniać dodatkowo rozgrywki.

- kolor, którym będą oznaczane kliknięte kwadraty (pod kluczem *mark_color*) – także zgodnie ze standardem CSS, zalecane jest podanie koloru nieumieszczonego w tablicy *square_colors*, aby zaznaczenie kwadratów było zawsze widoczne,
- rozmiar planszy (*board_size*) – liczba całkowita z przedziały od 5 do 12 włącznie (niespełnienie tego warunku spowoduje, że uruchomienie gry zakończy się błędem), określająca, z ilu kwadratów będzie się składał pojedynczy wiersz/kolumna naszej planszy. Im większy rozmiar, tym większe będzie okno interfejsu. Zalecany rozmiar to 10.

Brak pliku *config.json* lub niekompletny plik spowoduje, że uruchomienie gry nie będzie możliwe.

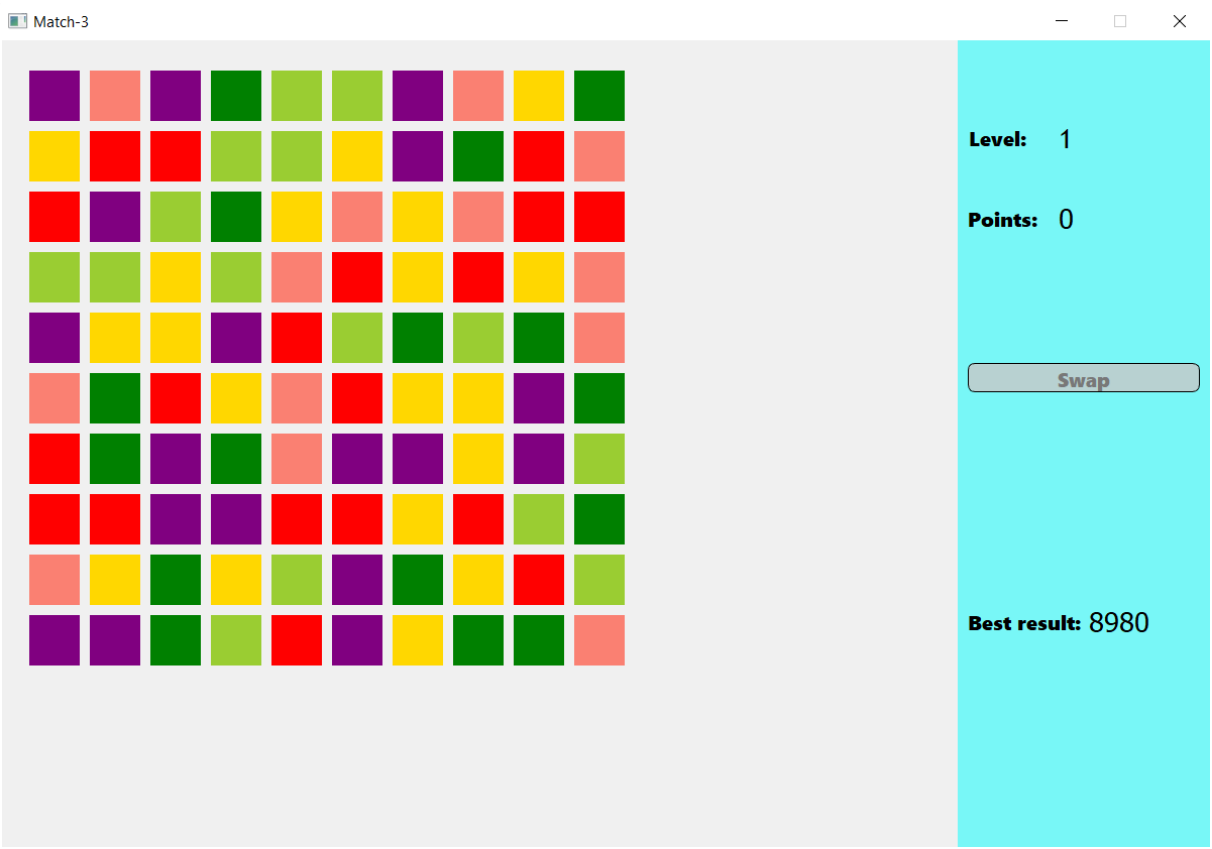


Przykładowa treść pliku *config.json*

Rozgrywka

Następnie można przystąpić do rozgrywki. W tym celu należy uruchomić w interpreterze plik *main.py*. Naszym oczom ukaże się interfejs z planszą po lewej stronie i turkusowym paskiem po prawej stronie. Na tym pasku znajdują się dane: aktualny poziom, aktualna liczba punktów, przycisk „Swap”, oraz informacja o najlepszym rezultacie (pobrana z pliku *result.yml*, jeśli pliku tego nie będzie, bo na przykład gra jest uruchamiana po raz pierwszy, w polu tym znajdzie się wartość 0).

Aby zamienić kwadraty miejscami należy kliknąć w dwa z nich, a następnie nacisnąć przycisk „Swap” – klikać można dowolne kwadraty na planszy (nie więcej niż dwa na raz), ale przycisk „Swap” będzie aktywny tylko wówczas, gdy kliknięte kwadraty po zamianie miejscami utworzą nam zbiór trzech lub więcej kwadratów w tym samym kolorze. Po kliknięciu w kwadrat pojawi się wokół niego 5-pikselowa obwolutka, która będzie takiego koloru, jak określono w pliku *config.json* (pod *mark_color*). Kwadrat można także odznaczyć, ponownie w niego klikając.



Wygląd interfejsu z planszą 10 x 10

Po kliknięciu przycisku „Swap” dwa kliknięte kwadraty zamienią się miejscami, następnie kwadraty tworzące trójkę/czwórkę itd. zostaną usunięte, a wszystkie kwadraty powyżej nich – spadną. W tym czasie klikanie innych przycisków nie będzie możliwe – gdy wszystkie kwadraty spadną, czasem będzie trzeba odczekać kilka sekund, zanim będzie można kontynuować rozgrywkę.

Awans

Po zdobyciu wystarczającej do awansu na kolejny poziom liczby punktów na ekranie pojawi się okno dialogowe informujące nas o tym fakcie. Aby móc je zamknąć i grać dalej, należy kliknąć przycisk „Ok”. Wtedy plansza zniknie i pojawi się nowa (w tym samym rozmiarze).

Przegrana

Gdy gracz nie będzie już miał żadnego możliwego ruchu, na ekranie pojawi się okno dialogowe informujące go, że przegrał, ile zdobył punktów. W oknie tym znajduje się pole tekstowe, do którego należy wprowadzić swoje imię (nick), a następnie kliknąć przycisk „Save result and quit” (przycisk ten będzie aktywny dopiero gdy w polu edycyjnym będzie co najmniej jeden znak). Gra się wówczas zamknie, a do pliku *result.yml* zapisane zostanie wprowadzone imię wraz z wynikiem (jeśli pliku nie ma, zostanie on utworzony).

3. STRUKTURA PROJEKTU

W katalogu głównym projektu znajdują się następujące pliki:

- *board.py*

Plik ten odpowiada za zasadniczy przebieg rozgrywki. Znajdują się w nim dwie zasadnicze klasy:

- Square – klasa, która definiuje pojedynczy kwadracik. Obiekt tej klasy posiada następujące atrybuty: współrzędne na planszy, napis warunkujący styl kwadratu (obramowanie i kolor tła), obiekt opisanej niżej klasy *Board*, do której należy.
- Board – plansza z kwadratami, klasa ta odpowiedzialna jest za logiczny i wizualny przebieg rozgrywki, m.in. za dodawanie punktów, animowane usuwanie kwadratów i dodawanie, wyświetlanie okien dialogowych, przeładowanie planszy po awansie na kolejny poziom
- Dwie dodatkowe klasy CustomAnimationGroup i CustomSequencialAnimationGroup – odpowiedzialne za animacje wywoływane przy ruchu kwadracików (ta pierwsza implikuje grupę wielu animacji pojawiających się jednocześnie, zaś druga – sekwencyjnie, jedna po drugiej), dziedziczą one po odpowiednich klasach z biblioteki PySide2

- *final_box.py*

Plik ten zawiera częściowo wygenerowany przez program QtDesigner kod, który generuje okienko wyskakujące podczas przegranej użytkownika. Znajduje się on w klasie LossBox.

- *main_window.py*

Inny plik częściowo wygenerowany przez QtDesigner. Zawiera on klasę Ui_MainWindow, która tworzy główne okno interfejsu gry – rozmiar tego okna będzie zależny od rozmiaru planszy.

- *main.py*

Plik odpowiedzialny za uruchomienie całej gry. Tworzy on instancje klas odpowiedzialnych za utworzenie okna interfejsu oraz planszy, wczytuje odpowiednie wartości z pliku konfiguracyjnego i pokazuje okno interfejsu na ekran.

- *read_from_config.py*

Plik zawiera funkcje wczytujące dane z pliku konfiguracyjnego i rzucające odpowiednie wyjątki, jeśli pliku nie ma lub zawiera niekompletne dane.

- *save_and_read_to_yml.py*

Zawiera funkcję zapisującą imię gracza i jego wynik do pliku *result.yml* i drugą odpowiedzialną za pobieranie danych z tego pliku i znalezienie najlepszego dotychczasowego wyniku.

- *sequences.py*

Posiada klasę Sequence, która pobiera 2-wymiarową kwadratową tablicę (w tym wypadku z kolorami kwadratów). Posiada dwie główne funkcje:

- wyszukującą trzy lub więcej takich samych elementów (kolorów) w jednej linii,
- znajdującą ruchy, które umożliwią takie ułożenie

W obu przypadkach funkcje te zwracają współrzędne kwadratów na planszy.

W folderze znajdują się także pliki zawierającymi testy jednostkowe do niektórych wyżej wymienionych klas (ich nazwy rozpoczynają się one od słowa *test*)