

# Projekt nr 13

## OBRAZKOWA UKŁADANKA

Tomasz Szkaradek  
Łukasz Wajda  
Monika Kidawska

### 1. Opis projektu

Celem projektu było napisanie gry komputerowej polegającej na ułożeniu obrazka z jego pomieszanych części. Program daje użytkownikowi możliwość wyboru jednej z dwóch wersji. Pierwsza zakłada przesuwanie kolejnych kwadratów w puste pole, natomiast druga polega na zamianie elementów obrazka ze sobą z równoczesnym ich odbiciem symetrycznym względem wybranej krawędzi.

Podczas rozgrywki mierzony jest czas oraz liczba kroków wykonanych przez gracza, a po zakończeniu dane te, przypisane do nicka grającego, umieszczane są w rankingu.

### 2. Założenia wstępne przyjęte w realizacji projektu

#### a) wymagania podstawowe

- W głównym oknie gry program losuje jeden z trzech przygotowanych wcześniej obrazków, dzieli go na daną liczbę elementów, miesza je i wyświetla na planszy.
- W pierwszej wersji gry użytkownik przesuwa kwadraty w puste pole, prowadząc do ułożenia obrazka.
- W drugiej wersji gry niektóre kwadraty są dodatkowo odbite symetrycznie względem różnych krawędzi. Użytkownik zamienia ze sobą dwa sąsiadujące elementy. Kliknięcie w ramkę między nimi powoduje również odbicie symetryczne względem wskazanej krawędzi.
- Po każdym ruchu sprawdzana jest poprawność ustawienia klocków, co sygnalizuje odpowiedni znak obok planszy.
- Program umożliwia wyświetlenie oryginalnego obrazka w trakcie gry po kliknięciu klawisza [Tab] na klawiaturze (lub przycisku „Podgląd” obok planszy). Ponowne jego kliknięcie powoduje powrót do wcześniejszego układu elementów.

#### b) wymagania rozszerzone

- Program wyświetla okno startowe, w którym użytkownik wpisuje swój nick oraz wybiera poziom trudności gry, czyli liczbę podziałów obrazka.
- Po kliknięciu przycisku Nowa Gra i wyświetleniu pomieszanych elementów obrazka zostaje uruchomiony licznik czasu oraz zliczanie liczby operacji.
- W pierwszej wersji gry kwadraty przemieszczają się w sposób płynny.

- W drugiej wersji gry kliknięcie na skrzyżowaniu krawędzi powoduje przesunięcie czterech sąsiadujących z nim kwadratów zgodnie z ruchem wskazówek zegara oraz ich odbicie symetryczne. Natomiast po kliknięciu w ramkę sąsiadującą z krawędzią planszy gry przemieszcza się cały wiersz lub cała kolumna.
- Zmiana kursora ze strzałki w rączkę sygnalizuje, czy kliknięcie we wskazane miejsce spowoduje ruch w grze.
- W trakcie rozgrywki gracz może zmienić swój nick, liczbę podziałów, wersję gry lub wylosować nowy obrazek.
- Osoba grająca ma również możliwość wyświetlenia instrukcji oraz informacji o grze.
- W momencie poprawnego ułożenia obrazka wyświetlany jest ranking dotychczas przeprowadzonych rozgrywek, w którym użytkownik ma możliwość wyboru sposobu sortowania (ze względu na nick, nazwę obrazka, liczbę podziałów, czas gry, liczbę kroków).

### 3. Analiza projektu

Interfejs użytkownika (ramka wraz z przyciskami i polami tekstowymi) zostały wykonane przy użyciu narzędzia wxFormBuilder.

Za pomocą przycisku Nowa Gra pobieramy obraz w formacie jpg – obraz jest przycinany do kwadratów lub prostokątów o rozmiarze określonym liczbą podziałów. Następnie wycięte i pomieszczone fragmenty obrazka można przesuwac według wybranej wersji gry. Użytkownik ma możliwość zapisu wyniku gry.

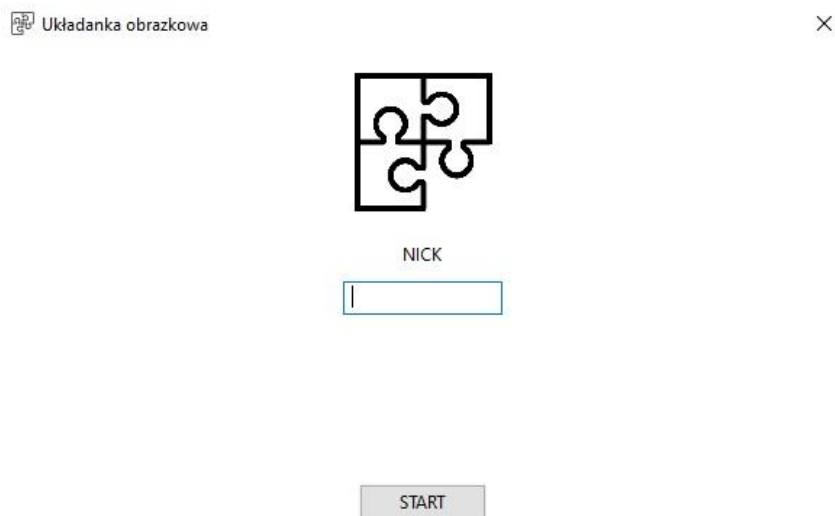
Dane przechowywane są w klasach biblioteki wxWidgets, głównie za pomocą wxBitmap i wxImage. Ilość podziałów jest wczytywana do programu na początku jako typ liczby całkowitej – „int”, podobnie jest również z trzema zmiennymi: liczbą sekund, minut oraz godzin jakie trwa określona gra.

Program komunikuje się z użytkownikiem za pomocą bocznego panelu, reprezentowanego przez elementy klasy GUIMyFrame1 opisujące działanie przycisków. Sama obsługa programu jest podzielona na kilka funkcji, obejmujących takie zagadnienia jak: wczytanie i zapis obrazka, reprezentowanie go jako pofragmentowane puzzle, zliczanie kroków i czasu, zapis do rankingu oraz wyświetlanie niezbędnych informacji.

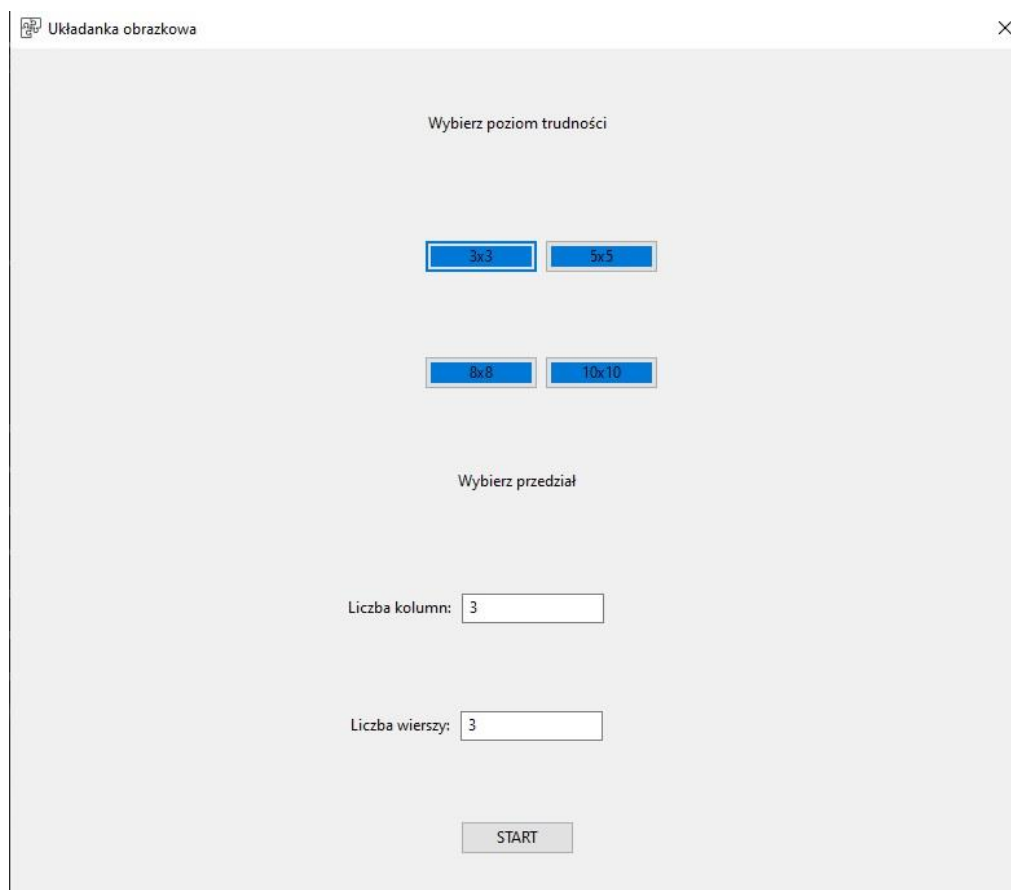
Cały projekt powstał w środowisku Visual Studio 2019, korzystając z standardowej biblioteki języka C++ (pliki nagłówkowe <iostream> i <cmath>), wxWidgets, która odpowiadała za obsługę środowiska graficznego oraz biblioteki FreeImage.

Poniżej przedstawione zostały poszczególne elementy interfejsu użytkownika:

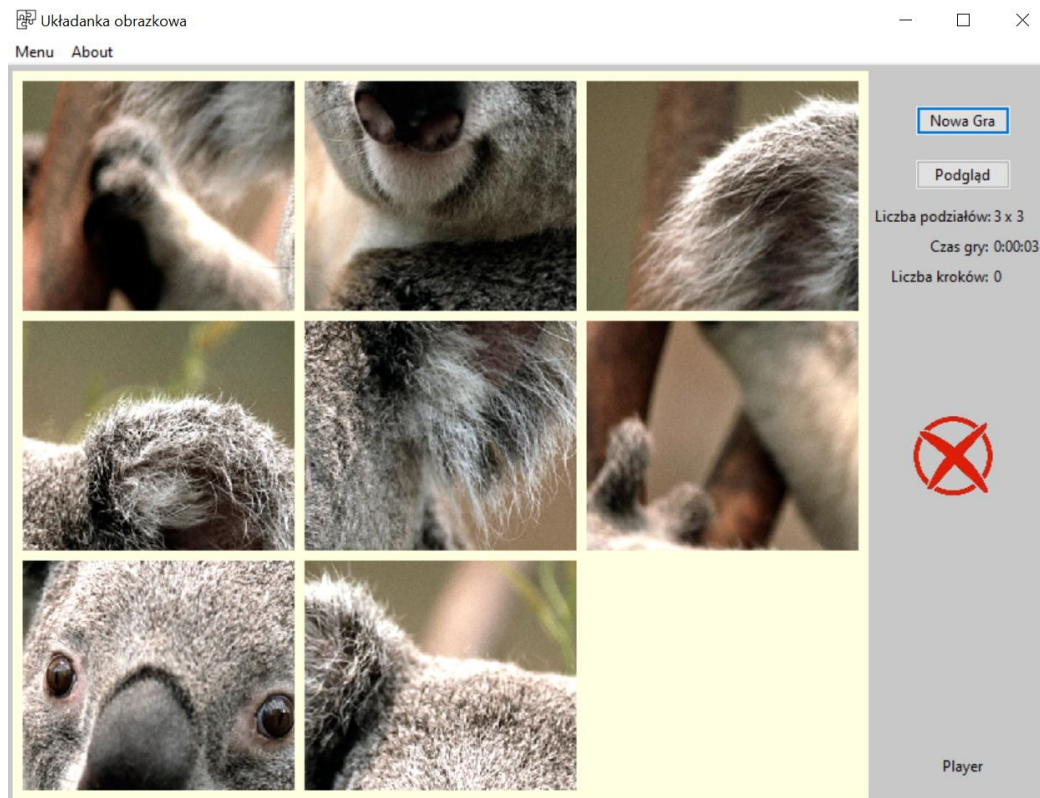
- Okno startowe programu:



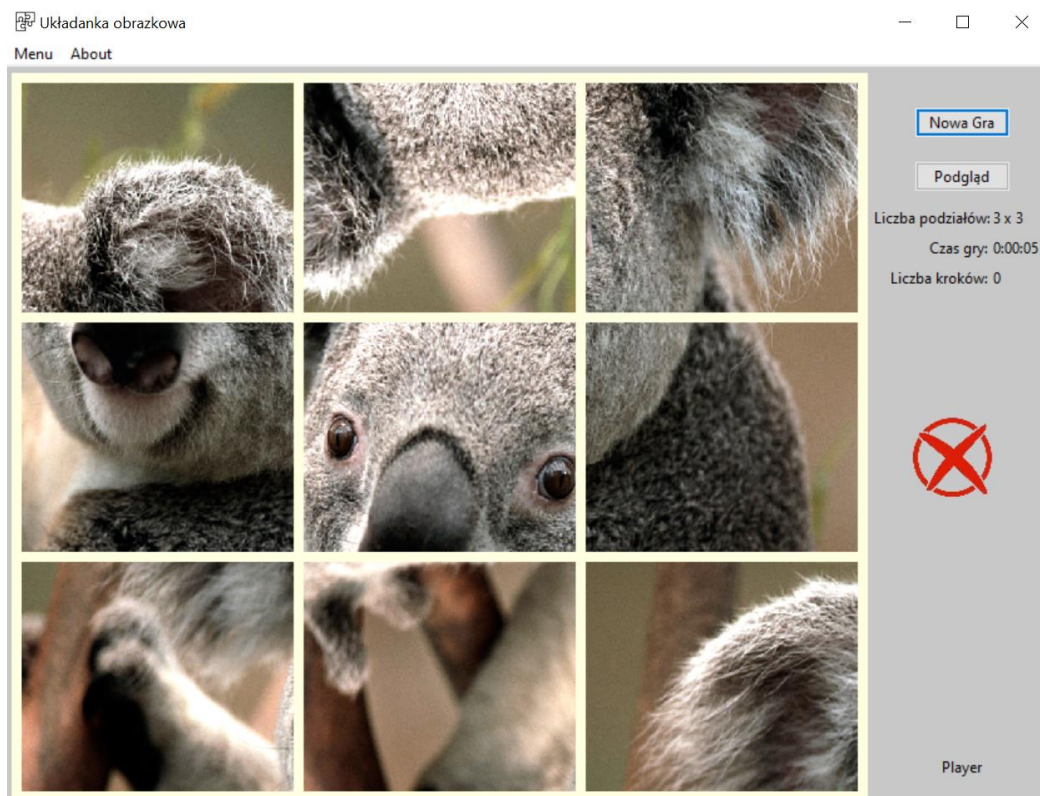
- Okno ustawień programu:



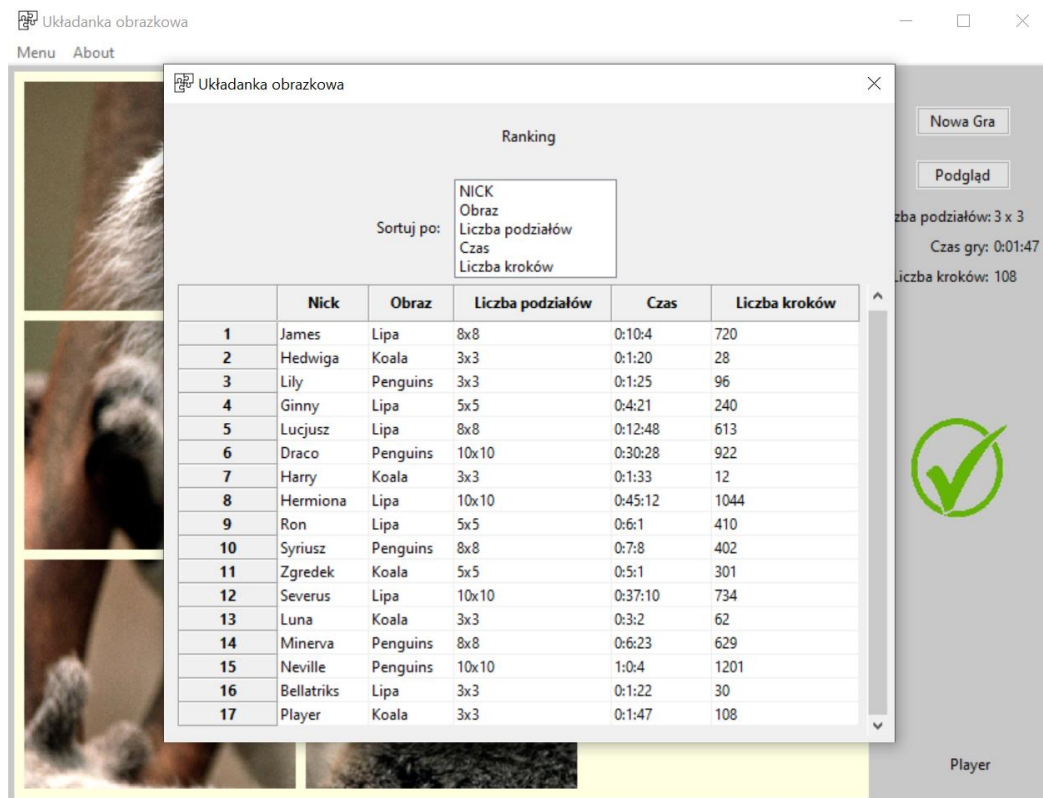
- Efekt działania programu dla 1 wersji gry:



- Efekt działania programu dla 2 wersji gry:



- Widok po pomyślnym zakończeniu gry (poprawnym ułożeniu obrazka):



Układanka obrazkowa

Menu About

Ranking

Sortuj po:

- NICK
- Obraz
- Liczba podziałów
- Czas
- Liczba kroków

	Nick	Obraz	Liczba podziałów	Czas	Liczba kroków
1	James	Lipa	8x8	0:10:4	720
2	Hedwiga	Koala	3x3	0:1:20	28
3	Lily	Penguins	3x3	0:1:25	96
4	Ginny	Lipa	5x5	0:4:21	240
5	Lucjusz	Lipa	8x8	0:12:48	613
6	Draco	Penguins	10x10	0:30:28	922
7	Harry	Koala	3x3	0:1:33	12
8	Hermiona	Lipa	10x10	0:45:12	1044
9	Ron	Lipa	5x5	0:6:1	410
10	Syriusz	Penguins	8x8	0:7:8	402
11	Zgredek	Koala	5x5	0:5:1	301
12	Severus	Lipa	10x10	0:37:10	734
13	Luna	Koala	3x3	0:3:2	62
14	Minerva	Penguins	8x8	0:6:23	629
15	Neville	Penguins	10x10	1:0:4	1201
16	Bellatriks	Lipa	3x3	0:1:22	30
17	Player	Koala	3x3	0:1:47	108

Nowa Gra

Podgląd

Liczba podziałów: 3 x 3

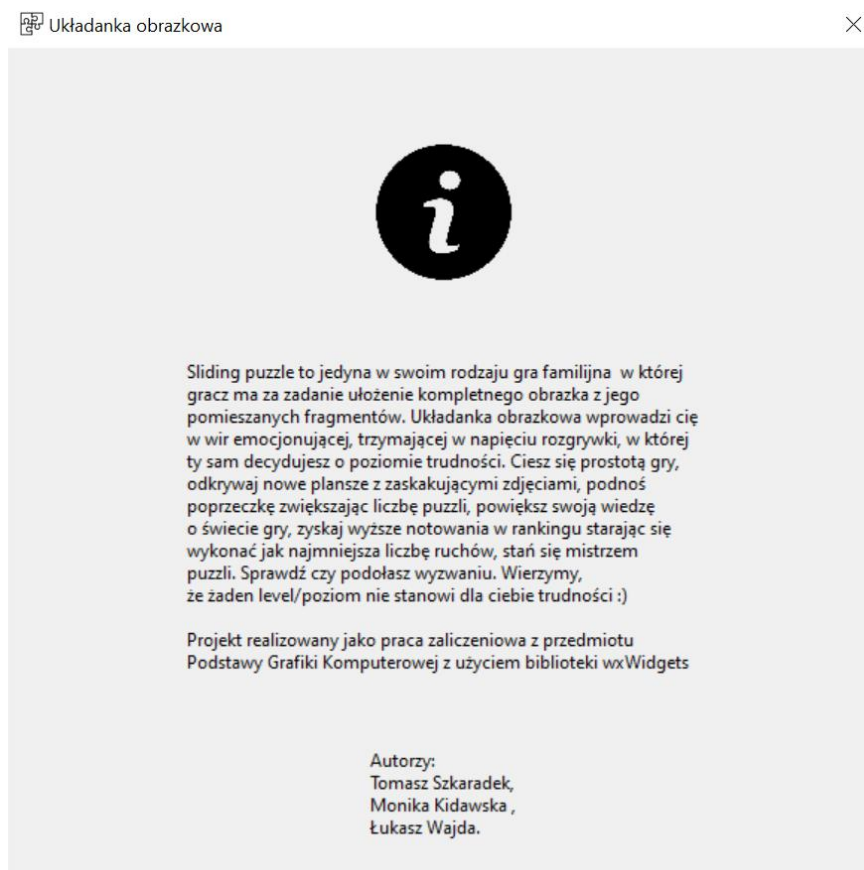
Czas gry: 0:01:47

Liczba kroków: 108

✓

Player

- Okno informacji o grze:



Układanka obrazkowa

×

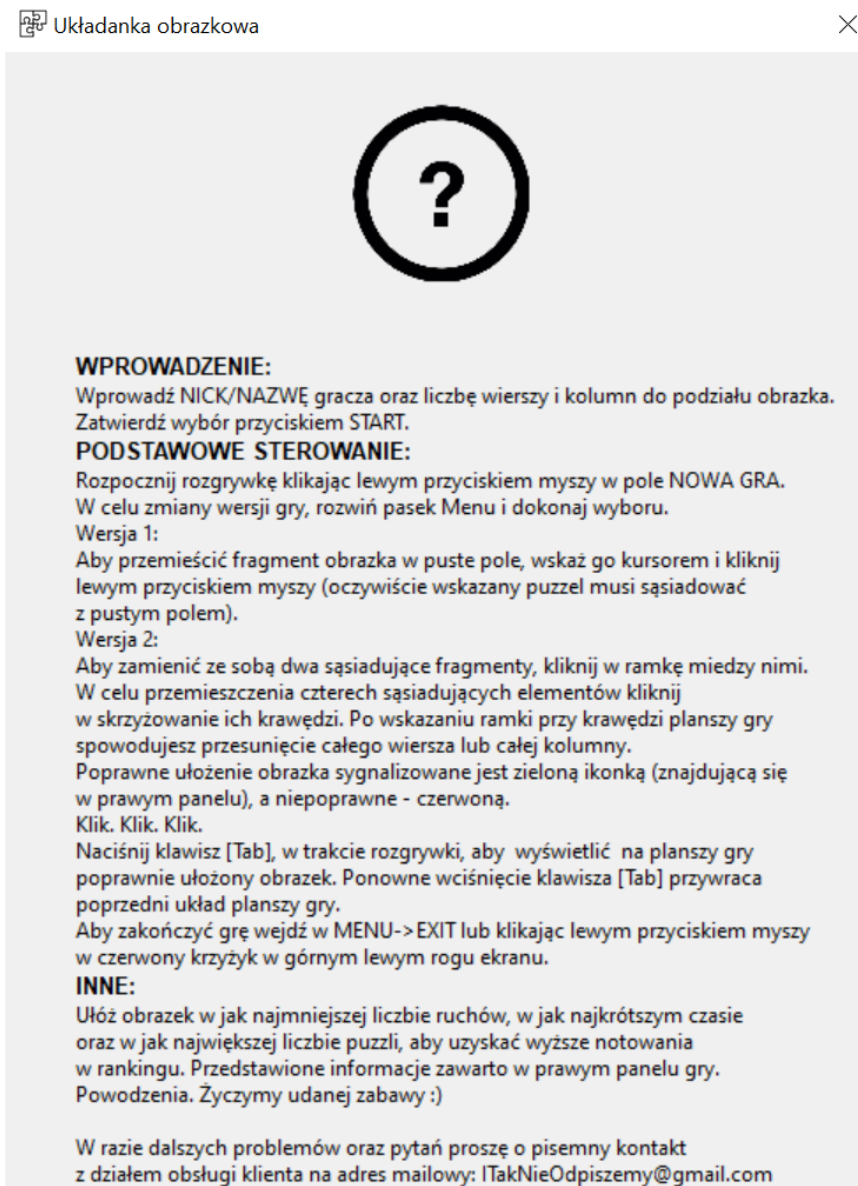
**i**

Sliding puzzle to jedyna w swoim rodzaju gra familijna w której gracz ma za zadanie ułożenie kompletnego obrazka z jego pomieszczonych fragmentów. Układanka obrazkowa wprowadzi cię w wir emocjonującej, trzymającej w napięciu rozgrywki, w której ty sam decydujesz o poziomie trudności. Ciesz się prostotą gry, odkrywaj nowe plansze z zaskakującymi zdjęciami, podnoś poprzeczkę zwiększając liczbę puzzli, powiększ swoją wiedzę o świecie gry, zyskaj wyższe notowania w rankingu starając się wykonać jak najmniejszą liczbę ruchów, stań się mistrzem puzzli. Sprawdź czy podołasz wyzwaniu. Wierzmy, że żaden level/poziom nie stanowi dla ciebie trudności :)

Projekt realizowany jako praca zaliczeniowa z przedmiotu Podstawy Grafiki Komputerowej z użyciem biblioteki wxWidgets

Autorzy:  
Tomasz Szkaradek,  
Monika Kidawska ,  
Łukasz Wajda.

- Okno wskazówek do gry:



#### 4. Podział pracy i analiza czasowa

Podczas realizacji projektu staraliśmy się wspólnie zastanawiać nad kolejnymi etapami oraz konsultować się na spotkaniach internetowych. Prace nad zadaniem rozpoczęły się dnia 13.05.2021r i trwały około trzy tygodnie. W tym czasie zrealizowaliśmy następujące działania:

- stworzenie szkicu interfejsu gry przy pomocy programu wxFormBuilder,
- wybór przykładowych obrazków, zaprogramowanie ich podziału, mieszania, zapisu do vectora oraz wyświetlania z ramkami na planszy gry,
- przygotowanie opisu gry oraz instrukcji dla gracza (okna „Informacje o grze” i „Help” w pasku „About”),



- stworzenie okna startowego umożliwiającego wpisanie nicku oraz wybór liczby wierszy i kolumn do podziału obrazka,
- zaprogramowanie zliczania czasu rozgrywki wraz z liczbą kroków wykonanych przez gracza,
- napisanie funkcji obsługującej kliknięcie w elementy obrazka oraz płynny ruch dla pierwszej wersji gry,
- umożliwienie sprawdzenia poprawnego ułożenia obrazka po kliknięciu klawisza [Tab] lub przycisku Podgląd,
- opracowanie drugiej wersji gry - odpowiednie mieszanie fragmentów obrazka, ich odbicia symetryczne oraz zamiana powiązana z kliknięciem poszczególnych krawędzi,
- utworzenie okna rankingu dotychczasowych rozgrywek,
- wprowadzenie sprawdzania warunku zakończenia gry dla obu wersji oraz wyświetlania rankingów i odpowiedniego znaku obok planszy gry,
- naniesienie ostatnich poprawek do kodu oraz przetestowanie działania całego programu,
- wykonanie dokumentacji.

## 5. Opracowanie i opis niezbędnych algorytmów

W naszym programie wykorzystaliśmy następujące algorytmy:

- algorytm podziału obrazka, zapisu go do vectora – przy użyciu dwóch pętli for pobieramy kolejne elementy obrazka przesuwając się wierszami o daną odległość wynikającą z wybranej przez użytkownika liczby wierszy i kolumn. Zapisujemy je kontenera typu vector, przechowującego obiekty klasy Picture. Dla pierwszej wersji gry ostatni element vectora stanowi puste pole. Przy przypisywaniu współrzędnych kolejnych kwadratów przyjmujemy także występowanie określonych odstępów między nimi dla lepszego ich wyodrębnienia na planszy gry.
- algorytm mieszania elementów oraz ich wyświetlania – wykorzystujemy oddzielną funkcję void Mix(), w której występują dwa różne sposoby mieszania kwadratów odpowiednio dla każdej wersji gry. Dla pierwszej z nich losujemy funkcją rand() elementy wyżej wspomnianego vectora, których współrzędne wymieniamy ze sobą. Natomiast dla drugiej wersji losujemy w pętli for zadaną przez zmienną sort\_degree kolejno krawędzie pionowe, poziome oraz skrzyżowania krawędzi. Następnie wywołujemy na wszystkich wylosowanych fragmentach funkcję Click1(). Zatem działamy „od tyłu” w celu zapewnienia możliwości ułożenia obrazka.  
Pomieszczone fragmenty obrazka wyświetlamy wierszami z wykorzystaniem wxBitmap oraz funkcji DrawBitmap wywoływanej na obiekcie wxClientDC.
- algorytm przemieszczania elementów na planszy – w pierwszej wersji gry sprawdzamy według współrzędnych, czy obok klikniętego elementu znajduje się puste pole. Jeśli się znajduje, zamieniamy współrzędne tych dwóch elementów, przesuwając płynnie kwadrat razem z fragmentem ramki, który

przylega do jego krawędzi leżącej naprzeciw kierunku ruchu (wyświetlamy kwadrat funkcją DrawBitmap w pętli while przemieszczając się o jeden piksel w odpowiednią stronę). W drugiej wersji gry sprawdzamy, czy wskazany fragment ramki występuje przy krawędzi planszy gry i wówczas przesuwany jest cały wiersz lub cała kolumna. W przeciwnym wypadku uwzględniamy trzy następujące zdarzenia:

- kliknięto ramkę poziomą - zamieniamy współrzędne dwóch sąsiadujących elementów w pionie, przy jednoczesnym ich odbiciu symetrycznym względem wskazanej krawędzi (poziomej),
  - kliknięto ramkę pionową - zamieniamy współrzędne dwóch sąsiadujących elementów w pionie, przy jednoczesnym ich odbiciu symetrycznym względem wskazanej krawędzi (pionowej),
  - kliknięto w skrzyżowanie ramek – zamieniamy współrzędne czterech sąsiadujących z nim elementów zgodnie z ruchem wskazówek zegara, przy jednoczesnym ich odbiciu symetrycznym względem odpowiedniej krawędzi.
- algorytm sprawdzania warunku zakończenia gry – po każdym wykonanym przez użytkownika ruchu wywoływana jest funkcja EndOfTheGame(), w której sprawdzamy w pętlach for, czy obecne współrzędne poszczególnych elementów zgadzają się z ich współrzędnymi sprzed przemieszania (czyli z prawidłowym ułożeniem obrazka). Dla drugiej wersji gry sprawdzamy dodatkowo, czy dany element nie jest odbity symetrycznie. Gdy napotkamy pierwsze błędne położenie, pętle zostają przerwane, natomiast pomyślimy przejściu pętli wywołujemy funkcję Win(), która zmienia znak obok planszy gry na zielony oraz wyświetla ranking.

## 6. Opis kodu

Opis programu, w tym szczegółowa dokumentacja klas, funkcji i zmiennych została zawarta w katalogu Doxygen.

## 7. Testowanie i wdrożenie

Osobnych testów wymagała przede wszystkim funkcja Draw. Okazało się, że kwestia dopasowania odstępów fragmenów obrazka jest zagadnieniem nietrywialnym. Wymagało to testowania wszystkich jej funkcjonalności oddzielnie od całego programu.

Problemy wystąpiły również podczas implementowania funkcji Click, nie wymagała ona jednak oddzielnych testów, jedynie drobnych poprawek w kodzie. Podczas pisania okazało się, że potrzebujemy krótszego czasu przejścia obrazka.

Problemy pojawiły się również z wydajnością – wymagały od nas zmiany sposobu przechodzenia fragmentów obrazka.



## **8. Wnioski**

Podstawowe i rozszerzone wymagania projektu zostały spełnione, jednak część z nich okazała się trudna w implementacji – szczególnie zagadnienia animacji przesuwania fragmentów obrazka. Istotna podczas pisania projektu była optymalizacja zagadnień oraz teoretyczne rozumienie zagadnienia. W przyszłości można było by popracować nad wyglądem gry.