

Krzyżówki

203.3

Prowadzący projektu: Krzysztof Gracki

Kindeusz Michał += Komarynska Daria += Majewska Aleksandra

Krzyżówka

- Krzyżówka – zagadka słowno-literowa polegająca na odgadywaniu lub dopasowywaniu haseł i wpisywaniu ich w krzyżujące się ze sobą rubryki. **Zadaniem projektu był program z generującymi się krzyżówkami.**

Zasady gry i cel

- Gracz za każde odgadnięte hasło otrzymuje punkty. Celem jest uzyskanie ich jak największej liczby i królowanie w rankingu krzyżówkowiczów.

Przebieg pracy nad projektem

- Na początku zdecydowaliśmy się na aplikację terminalową i skupiliśmy się na głównych funkcjonalnościach krzyżówki takich jak:
 - możliwość uzupełniania pól przez użytkownika
 - wyświetlanie pytań
 - sterowanie przebiegiem gry w terminalu

Oto efekt naszej początkowej pracy:

```
Welcome! Good luck! Have fun!

  1  2  3  4  5  6
1 | # | _ | # | _ | # |
2 | _ | _ | _ | _ | # |
3 | # | _ | _ | _ | _ |
4 | # | _ | _ | _ | _ |
5 | _ | _ | _ | _ | _ |

1. Działo (5, 1)
2. Zderzak pociagu (1, 2)
3. Co dzień inna w kalendarzu (2, 6)
4. Aktywowiec o l. a. 100 (2, 3)
5. Gospodarstwo hodowlane (3, 2)
6. Prawy dopływ Sekwany (1, 4)
7. Kaprał na statku (3, 5)
8. Złączka rur (2, 1)
9. Szata liturgiczna (4, 2)

1. FILL
2. END
```

```
1. FILL
2. END
1

1. QUESTION
2. FIELD
1
Enter number of question: 2
Enter answer: bufor
```

	1	2	3	4	5	6
1	#	b	#	-	#	#
2	-	u	-	-	#	-
3	#	f	-	-	-	-
4	#	o	-	-	-	-
5	-	r	-	-	-	-

1. Działo	(5, 1)
2. Zderzak pociagu	(1, 2)
3. Co dzien inna w kalendarzu	(2, 6)
4. Aktywowiec o l. a. 100	(2, 3)
5. Gospodarstwo hodowlane	(3, 2)
6. Prawy doplyw Sekwany	(1, 4)
7. Kapral na statku	(3, 5)
8. Zlaczka rur	(2, 1)
9. Szata liturgiczna	(4, 2)

1. FILL
2. END

Numery przy pytaniach to lokalizacje pierwszych liter odpowiedzi w formacie (rzad, kolumna). Użytkownik może uzupełnić krzyżówkę lub zakończyć grę. Po wybraniu opcji uzupełniania użytkownik decydował, czy chce podać całą odpowiedź, czy wypełnić konkretne pole. Po uzupełnieniu pojawiała się nowa plansza uzupełniona o odpowiedź.

Wersja terminalowa była mało podatna na błędy, ponieważ mogliśmy przetestować większość jej składowych. Jednak brakowało jej kilku ważnych funkcjonalności.

Po pierwsze nie było jeszcze podziału na hasła zorientowane poziomo i pionowo. Poza tym przy większych krzyżówkach znalezienie początku odpowiedniego hasła trwałoby zbyt długo. Dlatego zdecydowaliśmy się na wersję graficzną interfejsu użytkownika, która sama wyszuka odpowiednie pola, a gracz będzie mógł cieszyć się wygodą.

W międzyczasie trwały prace nad algorytmem tworzenia krzyżówek z listy słów.

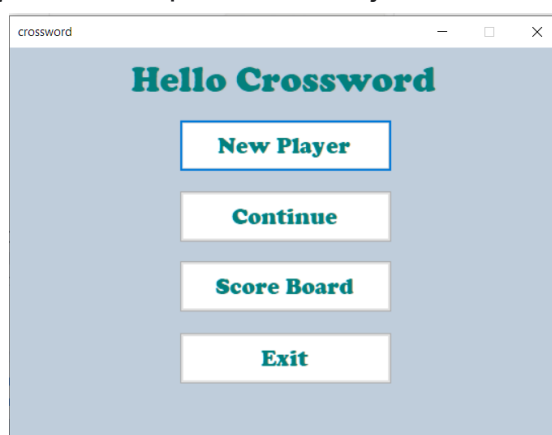
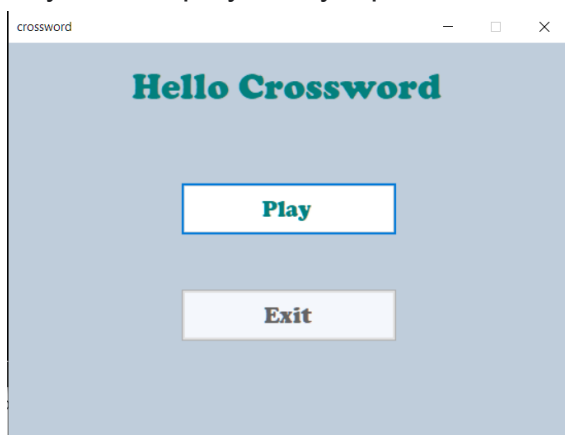
- Stworzenie modelu docelowego programu



Najpierw zdecydowaliśmy jak ma wyglądać interfejs gracza i zrobiliśmy wstępny szkic.

Pierwsze GUI wykonane było z użyciem **WinForms** w **Microsoft Visual Studio 2019**.

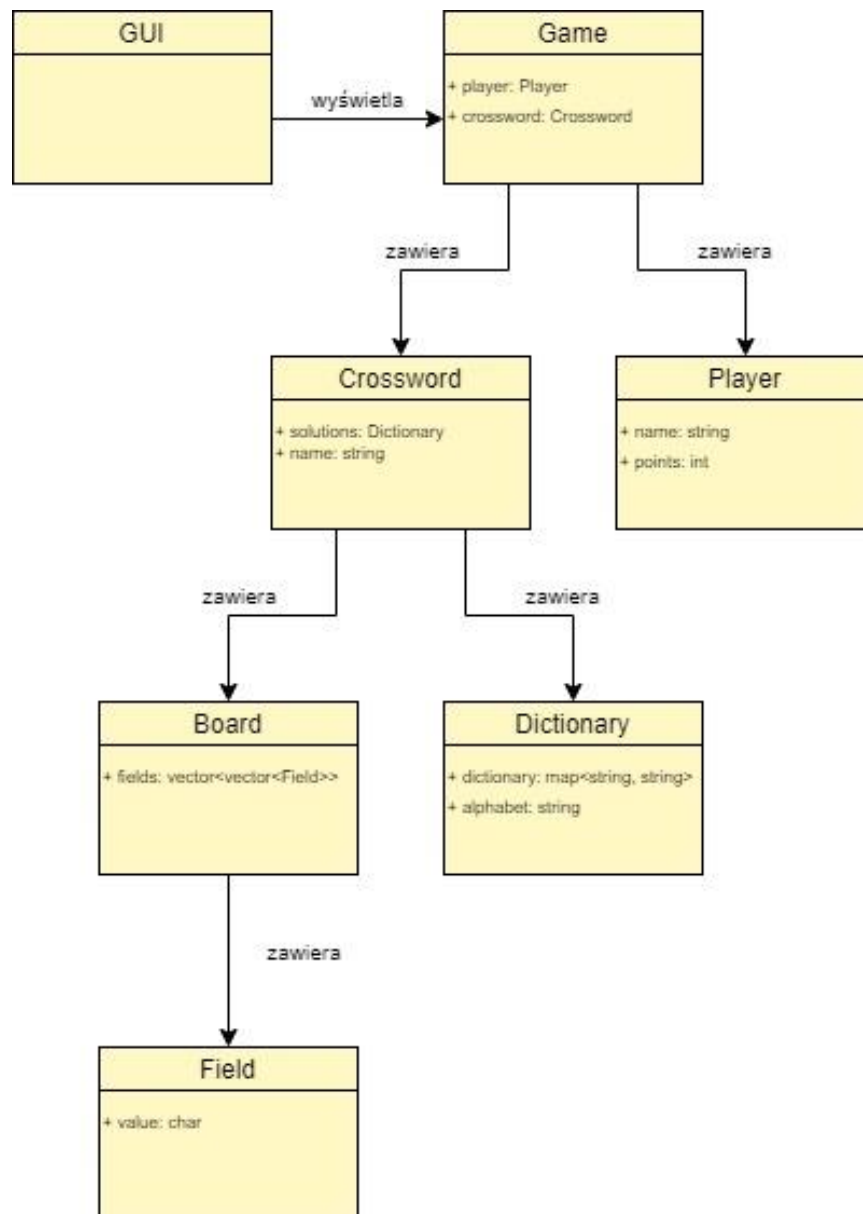
Jednak program szybko zrobił się zbyt skomplikowany i wprowadzenie niektórych funkcjonalności było niewykonalne przy naszym poziomie umiejętności. Oto pierwsza wersja GUI w WinForms:





- Po przeanalizowaniu wymagań i możliwości czasowych uznaliśmy, że zrealizujemy zaplanowany interfejs za pomocą **Qt**, który można podziwiać w sekcji *Opis działania*.
- W międzyczasie została zaimplementowana generacja krzyżówek możliwa również z poziomu GUI

Architektura projektu



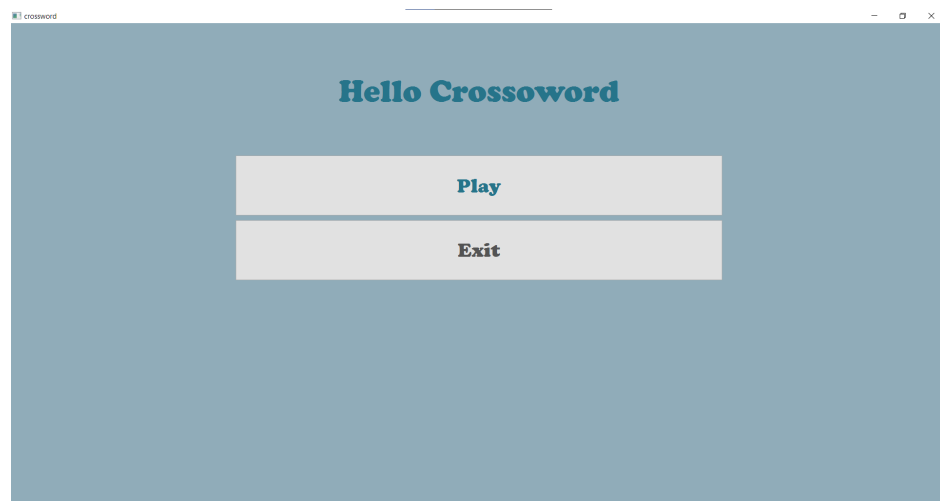
- *main.cpp* - plik jest przeznaczony do wywoływania uruchamiania programu, zawiera obiekt klasy Crossword.
- *Field.cpp* - plik zawiera klasę Field, przez którą można pracować z oddzielnym polem na planszy
- *Board.cpp* - plik zawiera klasę Board, która pozwala na stworzenie planszy i wypełnianie pól hasłami.
- *Crossword.cpp* - plik zawiera klasę Crossword, w której jest generowanie krzyżówki - wybór miejsca dla hasła i przyporządkowanie słów, przecinając je między sobą.
- *Player.cpp* - plik zawiera klasę Player, która umożliwia na przechowywanie danych o graczu, podczas i po rozgrywce.
- *Game.cpp* - plik zawiera klasę Game, w której jest uporządkowana kolejność rozgrywki.
- *word_list1.csv*, *crossword1.csv* - pliki zawierają przykładowe hasła z pytaniami, różnią się sposobem tworzenia, o których wspomniano było wyżej.
- Pliki tworzące GUI:
addcrosswordwindow.cpp, addplayerwindow.cpp, gamewindow.cpp, menuwindow.cpp, scoreboardwindow.cpp, selectcrossword.cpp, selectplayer.cpp, startwindow.cpp
oraz odpowiadające im pliki z rozszerzeniem .h, .ui

Wykorzystane narzędzia

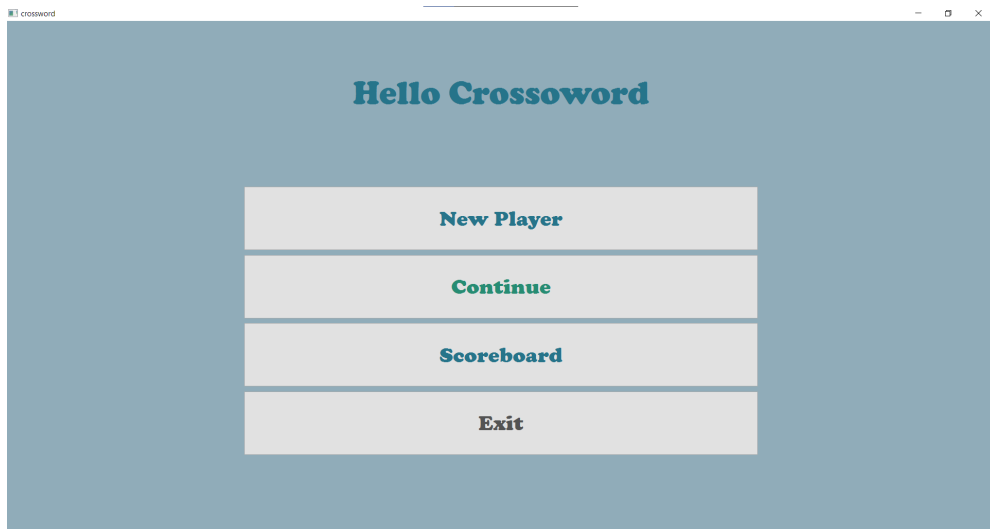
- Jedną z głównych technik wspomagających pracę było **TDD**(test-driven development). Testy jednostkowe pomogły nam w:
 - sprawdzeniu poprawności implementacji
 - planowaniu dalszych funkcjonalności
 - klarownym opisie działania funkcji
- Wykorzystując **GitLab**, planowaliśmy dalsze udoskonalenia i zwracaliśmy uwagę na problemy, tworząc **Issues**. Ułatwiło nam to:
 - dyskusję nad kodem
 - podział na podproblemy
 - podział pracy między uczestników
 - możliwość szybkiego zorientowania się w etapie pracy
- **Qt6** umożliwił szybkie tworzenie GUI dzięki **Qt designer**'owi i łatwej implementacji działania interfejsu przy użyciu **signals & slots**.
- **Microsoft Visual Studio 2019** ułatwiło nam debugowanie programu oraz sprawne pisanie testów jednostkowych przy użyciu **CppUnitTestFramework**.

Opis działania

Po uruchomieniu ukazuje się początkowe menu:



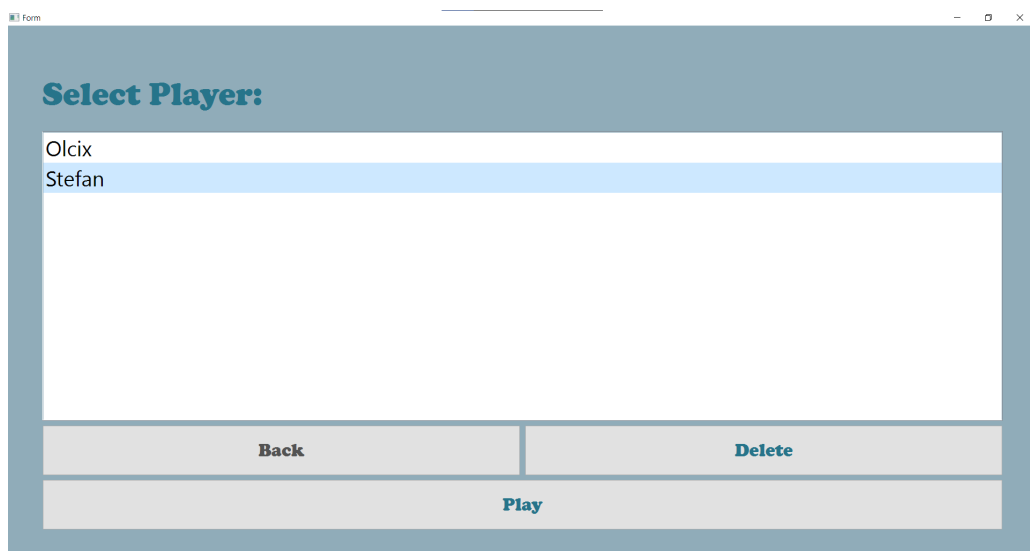
Nie ma w nim nic ekscytującego. Po wybraniu przycisku **Play** lądujemy w menu głównym, które przedstawia się następująco:



Można utworzyć nowego gracza wybierając **New Player**, kontynuować poprzednie rozgrywki, klikając **Continue** lub wyświetlić tabelę wyników.



Po wpisaniu nie użytej wcześniej i nie zawierającej przecinków nazwy użytkownika możemy go wybrać w menu wyboru gracza.



Po wybraniu możemy usunąć użytkownika (**Delete**) lub nim zagrać(**Play**). Po wybraniu opcji Play ukazuje nam się menu wyboru krzyżówki.

Player: **Stefan**

Select Crossword:

Moja gra

Buttons: Delete, Add, Logout, Play

Możemy utworzyć/dodać nową krzyżówkę, naciskając przycisk **Add**

Enter Name:

Enter File Name:

From list of words ☐

Buttons: Cancel, Add

Opcje generowania krzyżówki

- wczytanie gotowej krzyżówki z pliku **csv** o wierszach w formacie:
pytanie,odpowieź,rząd pierwszej litery,kolumna pierwszej litery,orientacja
w polu orientacja możliwe są opcje **vertically** lub **horizontally**
- utworzenie krzyżówki z listy pytań i odpowiedzi czytanej z pliku o wierszach w formacie
(wymagane jest zaznaczenie checkbox **From list of words**):
pytanie,odpowieź

Przy wyborze drugiej opcji można wielokrotnie używać tego samego pliku do generowania różnych krzyżówek z tej samej listy słów.

Po dodaniu nowa krzyżówka pojawi się na liście i można ją wybrać.

Przebieg gry

crossword

Moja gra

Player: Olcix

Score: 10 / 90

2

5

8

4

3

5

7

9

1

a

r

m

a

t

a

m

a

m

Horizontal:

1. Działo

5. Gospodarstwo hodowlane

8. Złaczka rur

9. Szata liturgiczna

Vertical:

2. Zderzak pociagu

3. Co dzień inna w kalendarzu

4. Aktywowiec o l. a. 100

6. Prawy dopływ Sekwany

7. Kapral na statku

Check

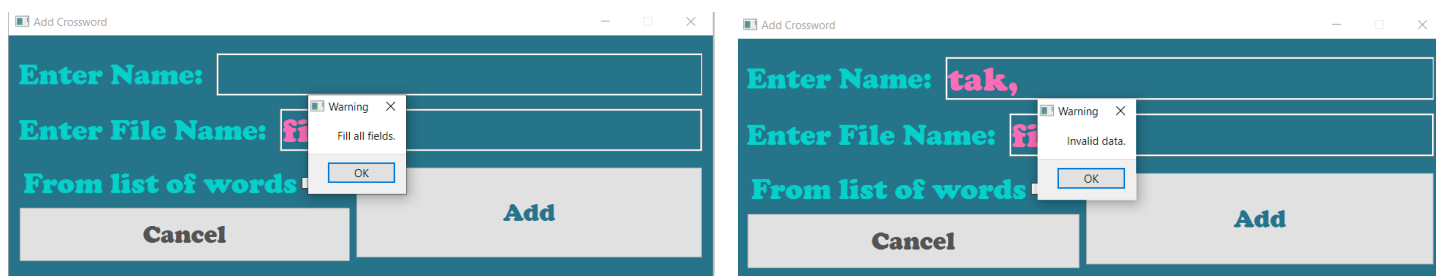
Finish

Na górze umieszczona jest nazwa bieżącej krzyżówki, imię gracza i jego zbierane punkty. Po prawej stronie znajduje się lista pytań, na które trzeba odpowiedzieć. Dla wygody na liście najpierw są umieszczone pytania do słów, które mają poziomą pozycję, a następnie pionową. Gdy użytkownik wybierze pytanie z listy zostaje ono podświetlone na . Gracz musi podawać swoją odpowiedź na pytanie, wpisując kolejne litery w odpowiednie dla danego słowa pola. W przypadku wpisania większej liczby liter brana pod uwagę będzie tylko pierwsza litera. Gdy użytkownik chce sprawdzić poprawność swoich odpowiedzi może to zrobić przez naciśnięcie klawisza **Check** w prawym dolnym rogu. Jeżeli całe wpisane hasło okaże się poprawne, użytkownik otrzyma **10 pkt.**. Będzie to widoczne w prawym górnym rogu. Ponadto pola z odgadniętym hasłem zostaną zablokowane (nie będzie możliwości edytowania) a literki podświetlą się na . Jeżeli gracz jest zadowolony z gry może zakończyć rozgrywkę naciskając **Finish** w prawym dolnym rogu. Jego wynik zostanie zapisany i doda się do sumarycznej liczby punktów, co gracz może podziwiać w **Scoreboard**:

Rank	Player	Score
1	Zomus	420
2	Daria	310
3	Michał	150
4	Mama	130
5	Olcix	90

Osobliwości

Przy łapaniu błędnych lub niepełnych danych użytkownik otrzymuje o tym odpowiednią informację.



Algorytm generowania krzyżówki

- Na początku program wybiera losową liczbę haseł od 3 do 15 z podanego pliku dla utworzenia krzyżówki. Następnie, również losowo wybiera się ta liczba słów.
- Przed układaniem, wszystkie słowa zostają posortowane:
 - według częstotliwości pojawiania się liter wśród wszystkich słów.
- Pierwsze słowo przyjmuje poziomą pozycję.
- Wyszukując wspólną literę dla hasła, wśród już posortowanych, sprawdzają się takie warunki:
 - czy poszczególne przewidywane pola są puste albo
 - czy znajduje się w polu kolejna wspólna litera,
 - czy potencjalna pozycja słowa nie będzie przedłużać wizualnie innego słowa na krzyżówce.
- Po sprawdzeniu warunków dla prawidłowej pozycji, hasło układa się na krzyżówkę:
 - w przeciwnym kierunku do słowa ze wspólną literą.

Uruchamianie programu

- Obowiązkowym jest instalowanie Qt 6. W przypadku pominięcia tego etapu nie uda się uruchomić programu, o co użytkownik będzie poinformowany przez komunikaty systemu. Jeżeli po zainstalowaniu Qt 6 komunikat dalej się pojawia, należy upewnić się, że ścieżka do wymaganych plików jest w zmiennej środowiskowej **PATH**.

crosswordGUI.exe - Błąd systemu

crosswordGUI.exe - Błąd systemu

×



Nie można kontynuować wykonywania kodu, ponieważ nie znaleziono obiektu Qt6Core.dll. Ponowne zainstalowanie programu może rozwiązać ten problem.

OK



Nie można kontynuować wykonywania kodu, ponieważ nie znaleziono obiektu libgcc_s_seh-1.dll. Ponowne zainstalowanie programu może rozwiązać ten problem.

OK

Dla uruchomienia aplikacji niezbędnym jest:

- Pobranie tego repozytorium
- Otwarcie folderu z plikami
- Uruchamianie gry z pliku crosswordGUI.exe

Podsumowując

Dzięki tej grze, można poszerzać wiedzę, rozwijać pamięć, myślenie logiczne i przestrzenne. Na tym etapie wszystkie ustalone przez zespół cele zostały wykonane. Całe skupienie zostało skierowane w stronę zaimplementowanego algorytmu generowania różnych krzyżówek i wygodnego i ciekawego interfejsu dla użytkownika. Dodatkowo istnieje możliwość rozbudowy projektu, wyróżniając różne poziomy trudności krzyżówek.

Słabe punkty i możliwe przyszłe ulepszenia

W trakcie implementowania zaplanowanych rozwiązań nasunęły nam się przemyślenia odnośnie
możliwej poprawy działania:

- Zabezpieczenie plików z danymi użytkowników i danymi krzyżówek

Na tym etapie skupiliśmy się na poprawności kodu, jednak aplikacja nadaje się jedynie do domowego użytku i zakłada, że użytkownik ma dobrą wolę i że na przykład nie doda sobie punktów, wpisując je do pliku.

- Zablokowanie możliwości robienia po raz kolejny tej samej krzyżówki

Gdy użytkownik zna już odpowiedzi może wygenerować dowolną ilość punktów tworząc wielokrotnie nową grę i odpowiadając na ponownie na te same pytania. Tutaj również zakłada się uczciwość użytkownika i jest to pole do poprawy.