

# Dokumentacja projektu Fiszki

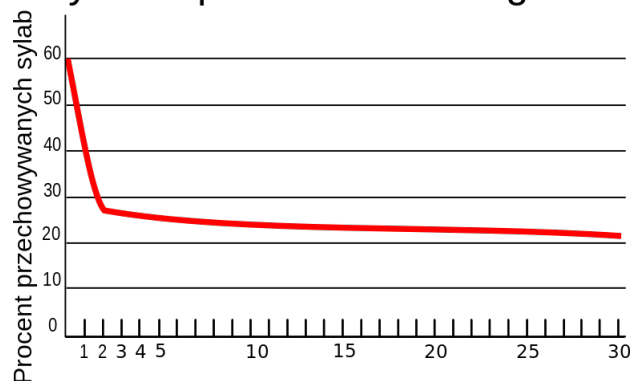
Marcin Kądziołka

29 czerwca 2021

## 1 Opis projektu

Nauka to walka z zapominaniem, dlatego należy powtarzać to, czego się nauczyliśmy. Nie należy jednak powtarzać zbyt często i wszystkiego, żeby nauka nie stała się nudna i męcząca. Potrzebny jest system nauki, który sprawi, że trudniejszy materiał będzie powtarzany odpowiednio często, by go zapamiętać, zaś materiał łatwiejszy — na tyle rzadko, by go nie zapomnieć. Zawodzi tutaj tradycyjna metoda nauki, czyli wielokrotne czytanie podręcznika. Dodatkowo zazwyczaj dzieje się to podczas jednej sesji, co sprawia, że w życie wchodzi krzywa zapominania. Jest to linia krzywa przedstawiająca według niemieckiego psychologa Hermanna Ebbinghausa zależność między ilością przechowywanej informacji w pamięci a upływem czasu, jaki nastąpił od momentu ich zapamiętania. Wygląda ona tak:

### Krzywa zapominania Ebbinghausa



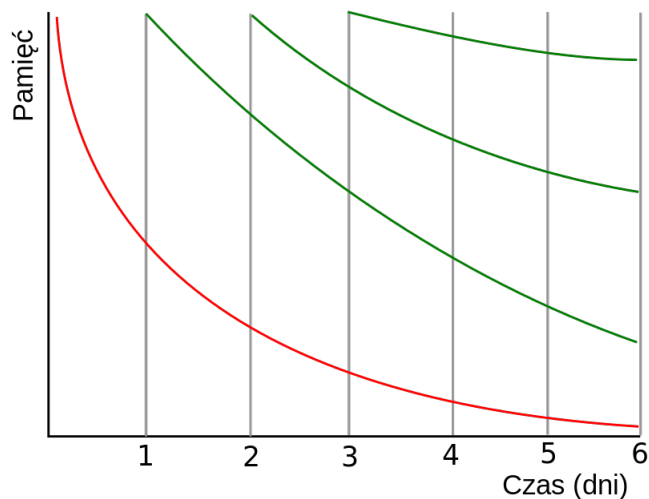
Jak widać, po zaledwie 2 dniach, pamiętamy zaledwie 20% powtarzanych informacji. Jest to wynik bardzo słaby i świetnie ukazujący wadę długiego uczenia się podczas jednej sesji. Skuteczniejszym rozwiązaniem są fiszki.

*Fiszka (fr. fiche) - kartka w kartotece dotycząca określonego hasła i zawierająca związane z nim informacje lub notatki.*

Fiszki można także wykorzystywać jako samodzielnie wykonaną pomoc naukową służącą do systematycznego uczenia się. Fiszka to kartonik z hasłem,

słówkiem, datą historyczną lub pytaniem na awersie oraz definicję, tłumaczeniem, objaśnieniem lub odpowiedzią na stronie odwrotnej (rewersie). Dzięki wyizolowaniu pojedynczych informacji, widać w którym zakresie mamy braki w wiedzy, a które tematy są nam doskonale znane. Pozwala to skupić się na częstszym powtarzaniu trudnego materiału. I właśnie to jest kluczem do lepszego zapamiętywania. Tym sposobem krzywa zapominania wygląda tak:

### Krzywa zapominania



Widać, że ze każdą powtórką, krzywa wypłaszcza się, co oznacza dłuższe przechowywanie informacji w pamięci.

Moim projektem jest przeniesienie tego analogowego systemu na aplikację komputerową. Taka wersja posiada znacznie więcej zalet i przeważa nad tradycyjną formą. Moja aplikacja pozwala na przechowywanie dowolnej liczby fiszek. Przechowywane na komputerze nigdy nie zgubią się, ani nie zniszczą. Ilość kategorii, w których chce się poszerzać wiedzę również jest nieograniczona.

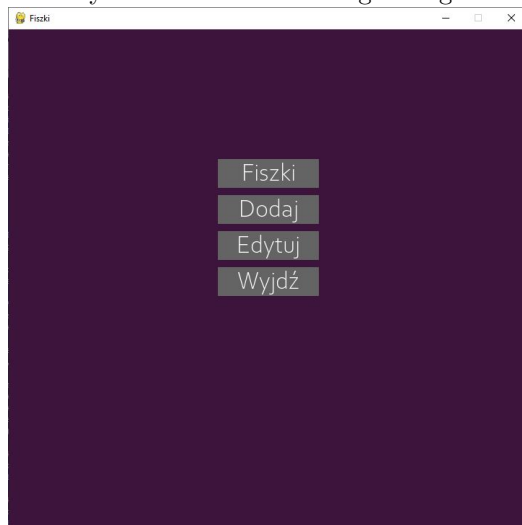
Działanie aplikacji opiera się na przechowywaniu danych (awersów i rewersów) w bazie, a następnie odpowiednim wyświetlaniu tych informacji użytkownikowi.

## 2 Procedura instalacji

Mój projekt korzysta z lokalnej bazy danych, więc aplikacja nie będzie działać uruchomiona na innym komputerze. Rozwiązaniem byłoby wykupienie serwera, aby aplikacja mogła się z nim łączyć z dowolnego urządzenia, na którym jest połączenie internetowe.

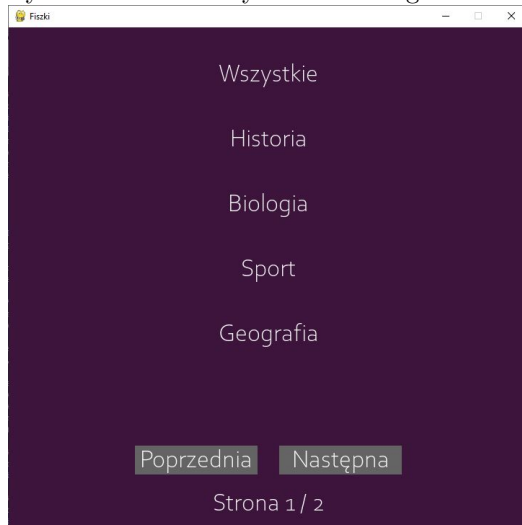
### 3 Wygląd i funkcjonalność aplikacji

Rysunek 1: Ekran menu głównego

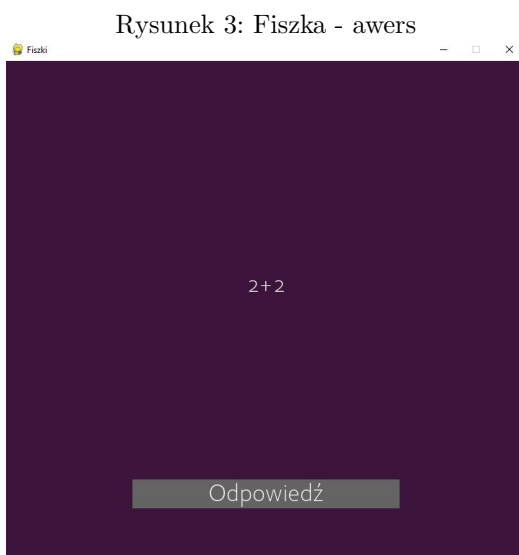


Pierwszy ekran, który widzimy po uruchomieniu aplikacji to menu główne. Stąd można wybierać dostępne funkcjonalności.

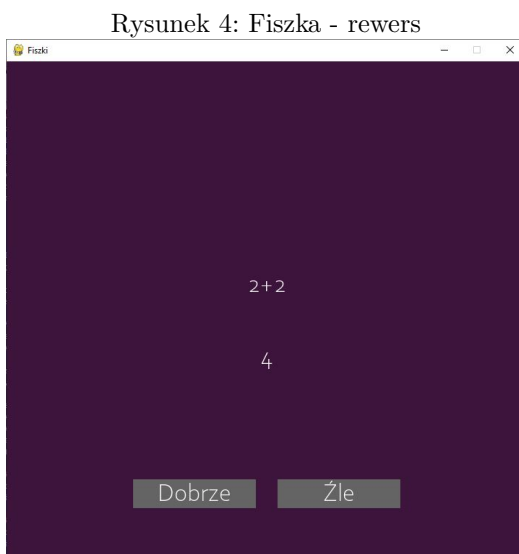
Rysunek 2: Ekran wybierania kategorii fiszek



Ten ekran pojawia się po kliknięciu w przycisk "Fiszki". Zanim zaczniemy się uczyć aplikacja pyta, czy mamy zamiar uczyć się z konkretnego zakresu.

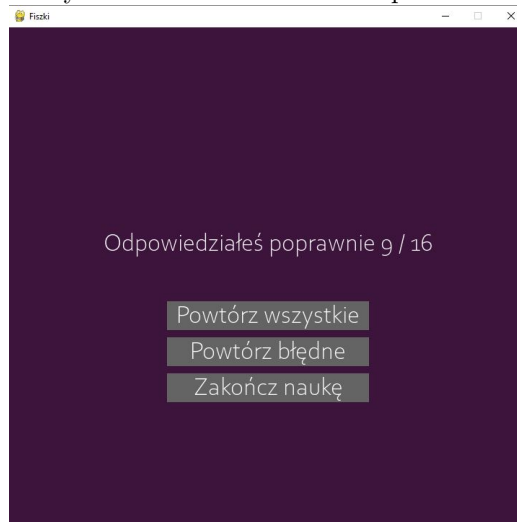


Po wybraniu kategorii, wyświetla nam się pierwsza fiszka. Teraz jest na czas na przypomnienie sobie odpowiedzi. Jeśli stwierdzimy, że taką znamy, klikamy "Odpowiedź".



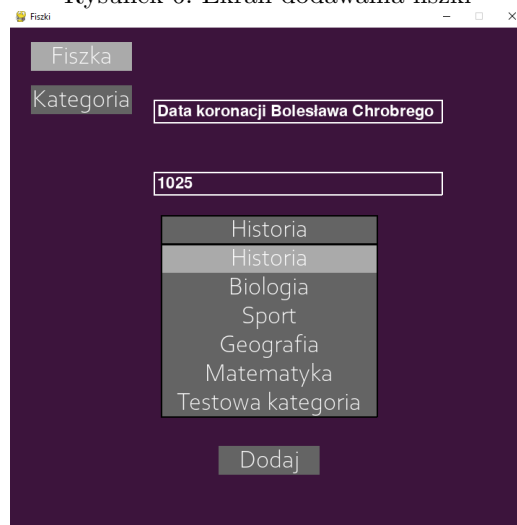
Teraz, porównujemy naszą odpowiedź z prawidłową i jeśli są one zgodne, wybieramy przycik "Dobrze", w przeciwnym wypadku wybieramy "Źle".

Rysunek 5: Podsumowanie odpowiedzi



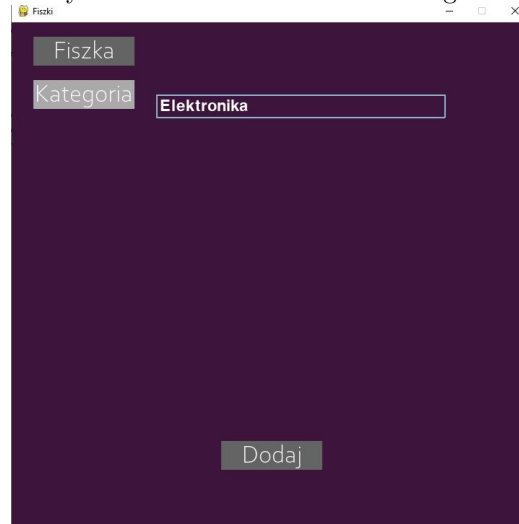
Po przejściu przez wszystkie fiszki z danej kategorii, widzimy ekran podsumowania i możemy zdecydować, czy osiągnięty rezultat należy poprawić.

Rysunek 6: Ekran dodawania fiszki



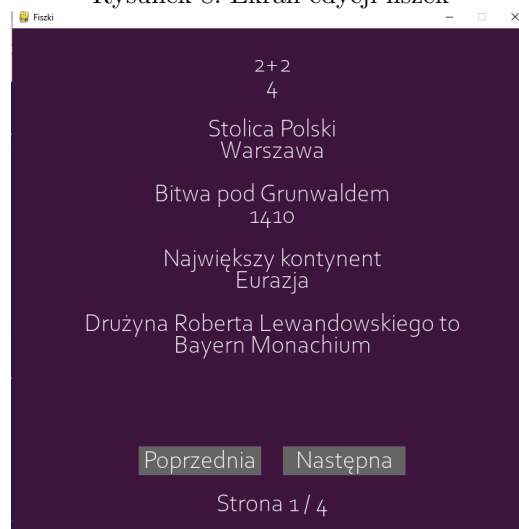
Po kliknięciu w menu głównym przycisku "Dodaj" ukazuje nam się ekran z polami tekstowymi. Pierwsze pole odpowiada za awers, drugie za rewers. Należy zaznaczyć z jakiej kategorii dodawana jest fiszka. Jeśli wszystko się zgadza klikamy "Dodaj".

Rysunek 7: Ekran dodawania kategorii



W opcji "Dodaj", możemy również dodać kategorię. Aby to zrobić należy zaznaczyć odpowiednie pole w lewym górnym rogu, wpisać nową kategorię i kliknąć "Dodaj".

Rysunek 8: Ekran edycji fiszek



Na ekranie edycji fiszek możemy wyświetlać wszystkie fiszki z bazy danych.

## 4 Użyte technologie i link do projektu

Do napisania aplikacji wybrałem język Python. To dynamiczny obiektowy język programistyczny, rozprowadzany na otwartej licencji umożliwiającej zastosowanie go do zamkniętych komercyjnych projektów. Python jest aktywnie rozwijany i posiada szerokie grono użytkowników na całym świecie i jest jednym z najpopularniejszych języków według różnego rodzaju metryk mierzących ilość tworzonych projektów, bibliotek i preferencji programistów. Jest to zatem język przyszłościowy i warto się w nim rozwijać. Jest to również język dla mnie najbardziej komfortowy, mam w nim największe doświadczenie. Dodatkowym atutem jest wiele przewodników, jak i dodatkowych bibliotek, czy narzędzi ułatwiających programowanie w tym języku.

Graficzny interfejs programu opiera się na bibliotece Pygame. To stworzona przez Pete Shinnersona biblioteka przeznaczona do tworzenia gier komputerowych oraz aplikacji multimedialnych w języku Python. Do działania wymaga biblioteki SDL, przy wykorzystaniu której dostarcza modułów pozwalających na wyświetlanie grafiki, odtwarzanie dźwięków, śledzenie czasu, obsługę myszy i joysticka, obsługę CD, czy renderowanie czcionek TTF. Pygame jako nakładka na SDL jest wieloplatformowa i umożliwia pracę na różnych systemach operacyjnych m.in. na Windows, Linux, MacOS. Biblioteka Pygame stanowi wolne oprogramowanie i jest dystrybuowana na zasadach licencji LGPL. Jej wybór również poprzedzony był moim wcześniejszym doświadczeniem z nią związanym.

Python oferuje łatwe połączenie z bazą danych MySQL. MySQL to wolnodostępny, otwartoźródłowy system zarządzania relacyjnymi bazami danych. Charakteryzuje się wygodnym i przejrzystym interfejsem. Kolejnym atutem jest szybkość działania.

Link do projektu:

<https://github.com/MarcinKadziolka/Projekt-inzynierski-Fiszki>

## 5 Napotkane problemy techniczne

Przy tworzeniu interfejsu graficznego, każde menu było osobną funkcją z własną pętlą, co powodowało, że przy przejściach między opcjami, nie zmieniała się pętla, tylko wchodziło się do następnej funkcji, a stara pętla się zatrzymywała, ale nie kończyła. Gdyby w ten sposób uruchamiać za każdym razem kolejne podmenu, doszłoby do zagnieżdżania się pętli w nieskończoność.

Rozwiązaniem okazało się postawienie warunku bool przy pętli, a każda funkcja podmenu po kliknięciu odpowiedniego przycisku zwraca wartość True lub False, co pozwala na przerwanie działania pętli i wrócenia "wyżej".

```
while running:
    #print("Fiszki")
    mouse = pygame.mouse.get_pos()
    click = False
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_ESCAPE:
                running = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1:
                click = True

    # colors the screen
    screen.fill(color_background)

    # Wypisujemy pytanie
    draw_text_on_screen(lista_fiszek[indeks_fiszek][1], width/2, height/2-50)

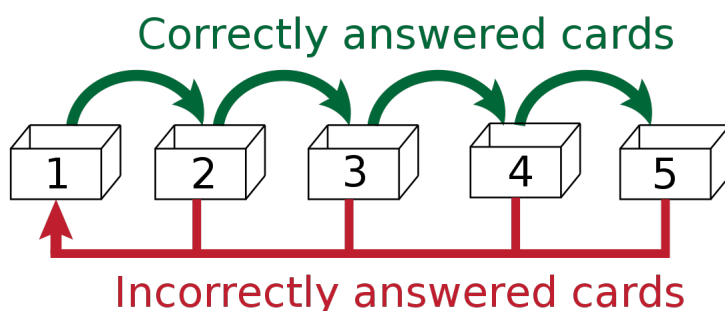
    draw_button(odpowiedz_button, color_dark)

    # Jeśli dotarliśmy do końca -> zerujemy indeks
    # Jeśli wciąż są fiszki do wyświetlania -> indeks++
    if odpowiedz_button.collidepoint(mouse):
        draw_button(odpowiedz_button, color_light)
        if click:
            running = odpowiedz()
```



## 6 Potencjalny plan rozwojowy na przyszłość

Podstawowym planem rozwojowym jest ukończenie wszystkich założonych funkcjonalności aplikacji, czyli wprowadzenie możliwości edycji istniejących fiszek i kategorii. Kolejnym krokiem byłoby opracowanie algorytmu decydującego, które z informacji należy powtórzyć w idealnym momencie tak, aby zminimalizować ilość powtórek, a zmaksymalizować ich efektywność. Standardową metodą jest korzystanie z podobnego schematu.



Polega on na stworzeniu kategorii częstotliwość powtarzania. Fiszki pierwszej kategorii, należy powtarzać codziennie, drugiej, co 2 dni, kategorii trzeciej, co 3 dni i tak dalej. Jeśli poprawnie odpowie się na daną fiszką, przenosi się ją do wyższej kategorii. Gdy nie udało się przypomnieć prawidłowej odpowiedzi, oznacza to, że informacja została zapomniana, więc musi wrócić na sam początek, aby utrwalić ją ponownie. Aby znaleźć odpowiednie przedziały czasowe, należałoby przeprowadzić badania na grupach kontrolnych, każda z różnymi przedziałami czasu powtarzania, a następnie porównać wyniki i użyć rozwiązania zastosowanego w grupie z najlepszymi osiągnięciami. Wtedy aplikacja byłaby usprawnionym odzwierciedleniem analogowej wersji fiszek.

Kolejnymi dodatkami aplikacji, zwiększającymi przewagę nad tradycyjnym system byłyby:

- możliwość dodawania zdjęć jako awers i rewers
- możliwość dodawania plików dźwiękowych jako awers i rewers
- prowadzenie i wyświetlanie różnorodnych statystyk