

# Narzędzia Pracy Grupowej – sprawozdanie z projektu

## Fiszki angielsko – polskie

### 1. Wykonawcy projektu

Mateusz Maż, [mateuszmaz@student.agh.edu.pl](mailto:mateuszmaz@student.agh.edu.pl)

Karolina Chochrek, [chochrek@student.agh.edu.pl](mailto:chochrek@student.agh.edu.pl)

Patryk Chwalczyk, [pchwalczyk@student.agh.edu.pl](mailto:pchwalczyk@student.agh.edu.pl)

### 2. Udostępnienie zasobów projektu

Aby zrealizować zadanie napisaliśmy program w języku Python oraz udostępniliśmy go w wirtualnym repozytorium na GitHubie: <https://github.com/Maziorow/fiszki>

Link do commitów: <https://github.com/Maziorow/fiszki/commits/master>

Link do Wiki na GitHub: <https://github.com/Maziorow/fiszki/wiki>

Link do pobrania projektu (gotowej aplikacji): <https://we.tl/t-oJgwOET4r9>

Aby rozdzielić zadania oraz śledzić postępy korzystaliśmy z narzędzia Trello: <https://trello.com/b/cS7C1mr0/fiszki>

### 3. Wstęp teoretyczny

Nasz projekt do nauki języka angielskiego został z tworzony z myślą o osobach, które chcą na bieżąco śledzić postępy w nauce oraz wybierać poziom trudności dostosowany do ich możliwości, więc wykorzystaliśmy do tego licznik czasu oraz zliczanie poprawnych odpowiedzi. Jedna sesja „gry” trwa do momentu aż odpowiemy poprawnie na wszystkie słowa. Jeżeli się pomylimy, słówko znowu przechodzi do bazy oczekujących w kolejce na odpowiedź. Dodaliśmy również opcję dodawania nowych słówek do bazy.

Jako interfejs graficzny użyliśmy kafelków oraz pól do wpisywania tekstu, dzięki temu aplikacja zachowuje prostotę i przejrzystość.

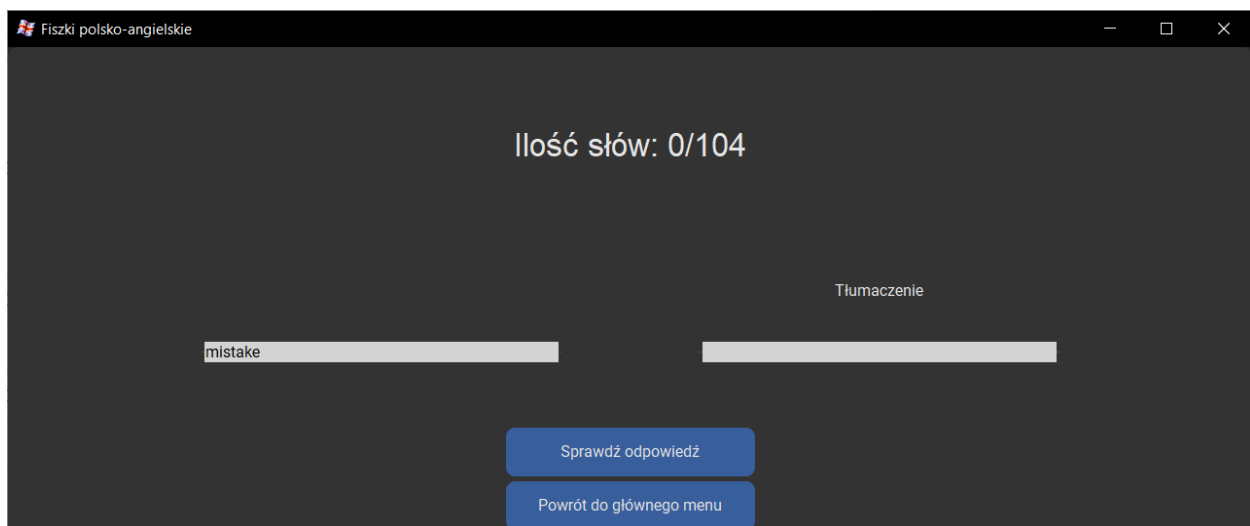
## 4. Podział obowiązków w grupie

### Mateusz Maź

Aspekt	Parametry	Wkład
Role	<p>Osoba nadzorująca grupę oraz wydająca polecenia co należy zrobić</p> <p>-przygotowanie środowiska graficznego pierwszej strony, na której będą odnośniki do kolejnych funkcjonalności</p> <p>-stworzenie funkcji zliczania czasu</p> <p>-udostępnienie programu na GitHub</p>	<p>Podjąłem się zadania zarządzania całą grupę moim pierwszymi zadaniami było stworzenie zadań w Trello następnie przystąpiłem do stworzenia odpowiedniego repozytorium w GitHub. Po wykonaniu zadań organizacyjnych przystąpiłem do tworzenia pierwszej strony graficznej w naszej aplikacji tak aby można było w łatwy sposób przechodzić pomiędzy funkcjonalnościami. Udostępniałem również wszystkie zmiany na GitHubie i dbałem o to, aby wszystkie zamieszczone tam programy były poprawne jak również pomagałem w pracy każdemu z członków zespołu.</p>
Kodowanie	Liczba linii kodu	214
	Funkcje (wymienić)	World, Sampleapp, PageOne
Repozytorium	Liczba commit-ów	6
	Liczba utworzonych gałęzi	1
	Gałąź (używana – nazwa)	Main
	Liczba dni aktywności GIT	6
Dokumentowanie	Liczba standup-ów	6
Aktywność	Liczba zrealizowanych zadań	Zrealizowane 4 zadania łącznie z dowodzeniem zespołem
	Szacowana liczba godzin	20
	Ocena procentowego wkładu	40%



Wygląd interfejsu po otwarciu pliku "Fiszki polsko-angielskie.exe"



Wygląd interfejsu po kliknięciu "Tryb nauki"

```

29 class Logic():
30     def __init__(self):
31         self.formatted_base = []
32         self.current_word = None
33         self.number_of_words = None
34         self.number_of_guessed = 0
35         self.number_of_mistakes = 0
36         self.words_base = pd.read_excel(db)
37         self.start_time = None
38         self.end_time = None
39         self.time_elapsed = None
40         self.learn_mode = None
41     def newBase(self, diff=None):
42         if diff == None:
43             for index, row in self.words_base.iterrows():
44                 self.formatted_base.append(Word(row['english'], row['polish'], row['difficulty']))
45         else:
46             for index, row in self.words_base.iterrows():
47                 if row['difficulty'] == diff:
48                     self.formatted_base.append(Word(row['english'], row['polish'], row['difficulty']))
49             self.number_of_words = len(self.formatted_base)

```

Zadanie zostało wykonane w dniu 06.04.2022r. było to jedno z pierwszych wykonanych zadań, ponieważ z pomocą tego interfejsu przechodziliśmy po kolejnych funkcjonalnościach kolejnym zdaniem było przygotowanie funkcji liczącej czas wykonywania ćwiczeń która była funkcjonalnością dodaną i wymyśloną przez nas jako funkcjonalność dodatkowa. Kod do niej znajduje się poniżej. Kolejnym ważnym zadaniem było stworzeni powrotów i odpowiednia komunikacja pomiędzy funkcjami tak aby była czytelna dla użytkownika co reprezentuje nam ostatni kod.

```
77     def getCounter(self):
78         return "Ilość słów: "+str(self.number_of_guessed)+"/"+str(self.number_of_words)
79     def startTimer(self):
80         self.start_time = time.time()
81     def stopTimer(self):
82         self.end_time = time.time()
83     def showTime(self):
84         sec = self.end_time - self.start_time
85         mins = sec // 60
86         sec = round(sec % 60, 2)
87         return ("{}:{1}".format(int(mins),sec))

90 class SampleApp(ctk.CTk):
91
92     def __init__(self, *args, **kwargs):
93         ctk.CTk.__init__(self, *args, **kwargs)
94         windll.shcore.SetProcessDpiAwareness(1)
95
96         self.title_font = tkfont.Font(family='Helvetica', size=72, weight="bold", slant="italic")
97
98         # the container is where we'll stack a bunch of frames
99         # on top of each other, then the one we want visible
100        # will be raised above the others
101        container = ctk.CTkFrame(self)
102        container.pack(side="top", fill="both", expand=True)
103        container.grid_rowconfigure(0, weight=1)
104        container.grid_columnconfigure(0, weight=1)
105
106        self.frames = {}
107        for F in (StartPage, PageOne, AddWordPage):
108            page_name = F.__name__
109            frame = F(parent=container, controller=self)
110            self.frames[page_name] = frame
111
112            # put all of the pages in the same location;
113            # the one on the top of the stacking order
114            # will be the one that is visible.
115            frame.grid(row=0, column=0, sticky="nsew")
116
117        self.show_frame("StartPage")
118
119    def show_frame(self, page_name):
120        '''Show a frame for the given page name'''
121        frame = self.frames[page_name]
122        frame.tkraise()
```

## Karolina Chochrek

Aspekt	Parametry	Wkład
Role	-zaprojektowanie estetycznego GUI  -przygotowanie bazy ponad 100 słówek polsko-angielskich  -przygotowanie funkcji dodawania własnych fiszek do bazy	Stworzenie interfejsu graficznego z pomocą Patryka przy użyciu Custom Tkinter, który znacznie wpłynął na wygląd i poprawił czytelność i estetykę.  Przygotowanie funkcjonalności, dzięki której użytkownik będzie mógł dodawać nowe słówka oraz przygotowanie bazy ponad 100 słówek.
Kodowanie	Liczba linii kodu	70
	Funkcje (klasy)	AddWordPage
Repozytorium	Liczba commit-ów	3
	Liczba utworzonych gałęzi	1
	Gałąź (używana – nazwa)	Main
	Liczba dni aktywności GIT	
Dokumentowanie	Liczba standup-ów	6
Aktywność	Liczba zrealizowanych zadań	Zrealizowane 3 zadania
	Szacowana liczba godzin	12
	Ocena procentowego wkładu	30%

	A	B	C	D
1	po angielsku	po polsku		poziom trudności
2	activity	aktywność	działania	łatwy
3	accident	wypadek	awaria	łatwy
4	actress	aktorka	artystka	łatwy
5	after	później	za	łatwy
6	adventure	przygoda	odważyć się na coś	łatwy
7	advice	rada	wskazówka	łatwy
8	toy	zabawka	bawić się czymś	łatwy
9	towel	ręcznik	wycierać się ręcznikiem	łatwy
10	tourist	turysta	zwiedzający	łatwy
11	try	próbować	próba	łatwy
12	trip	wycieczka	podróż	łatwy
13	uniform	mundur	jednolity	łatwy
14	uncle	wujek	stryj	łatwy
15	walk	spacer	isła	łatwy
80	longbow	łuk	łucznik	trudne
81	dagger	sztylet	nóż	trudne
82	figures	liczby	postaci	trudne
83	dividend	dywidenda	wypłaty	trudne
84	commodity	towar	dobro	trudne
85	hemisphere	półkula	hemisfera	trudne
86	vanity	toaletka	próżność	trudne
87	subdued	przygnębiony	obezwładniony	trudne
88	vivacious	pełen życia	skoczny	trudne
89	exuberant	entuzjastyczny	żywiłowy	trudne
90	closeness	bliskość	towarzystwo	trudne
91	anguish	cierpienie	udręka	trudne
92	gilet	kamizelka	bezrękawnik	trudne
93	rectory	plebania	probostwo	trudne
94	chalet	domek w górach	schronisko	trudne
95	abbey	opactwo	klasztor	trudne
96	motherboard	płyta główna		trudne
97	encryption	szyfrowanie	szyfr	trudne
98	peacock	paw		trudne
99	lynx	ryś		trudne
100	carnivore	mięsożerca (osoba)	mięsożerca (zwierzę)	trudne
101	dove	gołąb		trudne

*Baza słówek przechowywana w Microsoft Excel*

Fizyki polsko-angielskie

Słowo po polsku

Słowo po angielsku

Poziom trudności

Dodaj słowo

Powrót do głównego menu

*Wygląd interfejsu po kliknięciu "Dodaj słowo"*

```

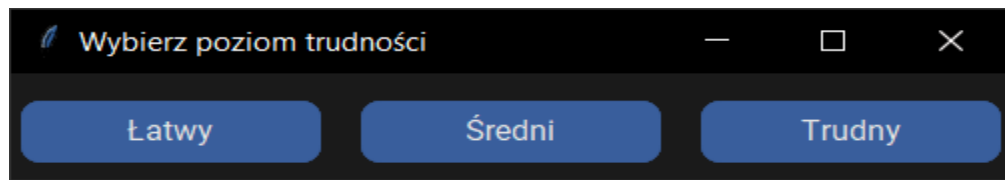
346     def add_word(pl,en,diff):
347         if pl.get() and en.get() and diff.get():
348             try:
349                 new_word = pd.DataFrame([{'polish':pl.get(), 'english':en.get(), 'difficulty':diff.get()}])
350                 wb = load_workbook(filename = "baza_slowek_polsko_angielskie.xlsx")
351                 ws = wb["Arkusz1"]
352                 for r in dataframe_to_rows(new_word, index=False, header=False):
353                     ws.append(r)
354                 wb.save("baza_slowek_polsko_angielskie.xlsx")
355                 pl.delete("0", 'end')
356                 pl.insert(ctk.END,"")
357                 en.delete("0", 'end')
358                 en.insert(ctk.END,"")
359                 diff.set('')
360             def createToplevel():
361                 self.window = ctk.CTkToplevel(self)
362                 self.window.grab_set()
363                 self.window.geometry("400x200")
364                 self.window.title("")
365
366                 napis = ctk.CTkLabel(self.window, text="Dodano słowo")
367                 napis.place(relx=0.15, rely=0.2, relheight=0.1, relwidth=0.7)
368
369                 button_medium = ctk.CTkButton(self.window, text="Kontynuuj",
370                                             command=lambda: closeToplevel())
371                 button_medium.place(relx=0.15, rely=0.82, relheight=0.1, relwidth=0.7)
372
373             def closeToplevel():
374                 self.window.destroy()
375
376             createToplevel()

```

Funkcjonalności przedstawione na powyższych zrzutach ekranów to w pierwszej kolejności baza słówek wykonana w programie Excel do dnia 18.04.2022r. Kolejną częścią jest opcja dodawania słówek przez użytkownika do istniejącej już bazy, której działanie jest zrealizowane powyższym kodem z datą wykonania 13.05.2022r.

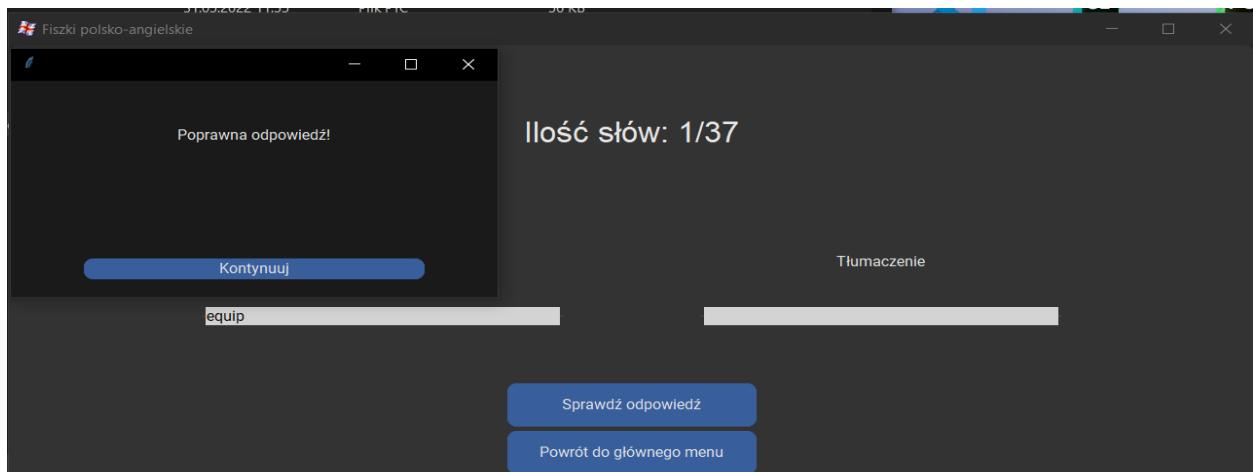
## Patryk Chwalczyk

Aspekt	Parametry	Wkład
Role	-zaprojektowanie estetycznego GUI  -stworzenie poziomów nauki, w których będzie można sprawdzać postęp	Stworzenie interfejsu graficznego z pomocą Karoliny przy użyciu Custom Tkinter, który znacznie wpłynął na wygląd i poprawił czytelność i estetykę. Funkcjonalnością, którą stworzyłem z pomocą Mateusza ma za zadanie umożliwić użytkownikowi wybrać poziom nauki przy sprawdzeniu oraz sprawdzić czas poświęcony na naukę danych fiszek.
Kodowanie	Liczba linii kodu	60
	Funkcje (wymienić)	StartPage  chooseDifficulty
Repozytorium	Liczba commit-ów	3
	Liczba utworzonych gałęzi	1
	Gałąź (używana – nazwa)	Main
	Liczba dni aktywności GIT	3
Dokumentowanie	Liczba standup-ów	6
Aktywność	Liczba zrealizowanych zadań	Zrealizowane dwa zadania
	Szacowana liczba godzin	12
	Ocena procentowego wkładu	30%



Wygląd interfejsu po kliknięciu "Sprawdź swoją wiedzę"





*Wygląd interfejsu po wpisaniu poprawnej odpowiedzi*

```

132     # title and icon of the application
133     controller.title("Fiszki polsko-angielskie")
134     controller.iconbitmap("pictures/xp_uk.ico")
135
136     window_width = 1024
137     window_height = 400
138     screen_width = controller.wininfo_screenwidth()
139     screen_height = controller.wininfo_screenheight()
140     center_x = int(screen_width / 2 - window_width / 2)
141     center_y = int(screen_height / 2 - window_height / 2)
142     controller.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')
143
144     label = ctk.CTkLabel(self, text="Fiszki polsko-angielskie", text_font=("Bank Gothic Medium BT", 40))
145     label.place(relx=0.1, rely=0.1, relheight=0.1, relwidth=0.8)
146
147     button1 = ctk.CTkButton(self, text="Tryb nauki",
148                             command=lambda: chooseDifficulty())
149     button1.place(relx=0.01, rely=0.5, relheight=0.1, relwidth=0.3)
150
151     button2 = ctk.CTkButton(self, text="Dodaj słowo",
152                             command=lambda: controller.show_frame("AddWordPage"))
153     button2.place(relx=0.69, rely=0.5, relheight=0.1, relwidth=0.3)
154
155     button3 = ctk.CTkButton(self, text="Sprawdź swoją wiedzę",
156                             command=lambda: createToplevel())
157     button3.place(relx=0.35, rely=0.5, relheight=0.1, relwidth=0.3)
158
159     exit_button = ctk.CTkButton(self, text="Zamknij aplikację", command=lambda: exit())
160     exit_button.place(relx=0.35, rely=0.75, relheight=0.1, relwidth=0.3)

```

Zadaniem było stworzenie i obrobienie interfejsu graficznego tak aby wglądał schludnie i ładnie. Wykorzystana do tego została biblioteka Custom Tinker, której użycie reprezentuje powyższy kod. Zostało to dokończzone jako jedno z ostatnich zadań dnia 24.05.2022r. Wcześniejszym zadaniem było przygotowanie funkcjonalności dotyczącej trybów nauki, których kod znajduje się poniżej.

```

180     def chooseDifficulty(diff=None, learn_mode=True):
181         logic.newBase(diff)
182         logic.randNewWord()
183         logic.learn_mode = learn_mode
184         app.frames["PageOne"].T1.config(state='normal')
185         app.frames["PageOne"].T1.delete("0", "end")
186         app.frames["PageOne"].T1.insert(ctk.END, logic.getCurrentWord())
187         app.frames["PageOne"].T1.config(state='disable')
188         app.frames["PageOne"].counter.config(text=logic.getCounter())
189         logic.startTimer()
190         app.show_frame("PageOne")

```

## 5. Wnioski

Projekt był dla nas ciekawą przygodą, dzięki której nauczyliśmy się dobrze współpracować z wykorzystaniem aplikacji takich jak Trello w której zarządzaliśmy powierzonymi nam zadaniami i sprawdzaliśmy postęp prac. Cały nasz kod został umieszczony w repozytorium na GitHubie, gdzie mógł zostać sprawdzony przez pozostałych członków zespołu i umożliwiał stworzenie dokumentacji do naszej pracy. Niestety nie obyło się bez trudności, ponieważ z racji tego, że był to dla większości z nas pierwszy tak duży projekt w Pythonie musieliśmy nauczyć się korzystania z Pycharma w sposób oparty na obiektach i strukturach oraz korzystania z odpowiednich bibliotek dzięki czemu mogliśmy znacznie usprawnić i przyspieszyć naszą pracę. Wynikły też problemy z samym GitHubem, ponieważ to również był pierwszy nasz projekt, który w pełni korzystaliśmy z repozytorium, jednakże z problemów wynikających z kolizji z pracy jednocześniej staraliśmy się znacznie lepiej przygotować program tak aby na zdalnym repozytorium nie musieć nic zmieniać.