

Część I. Aplikacja internetowa

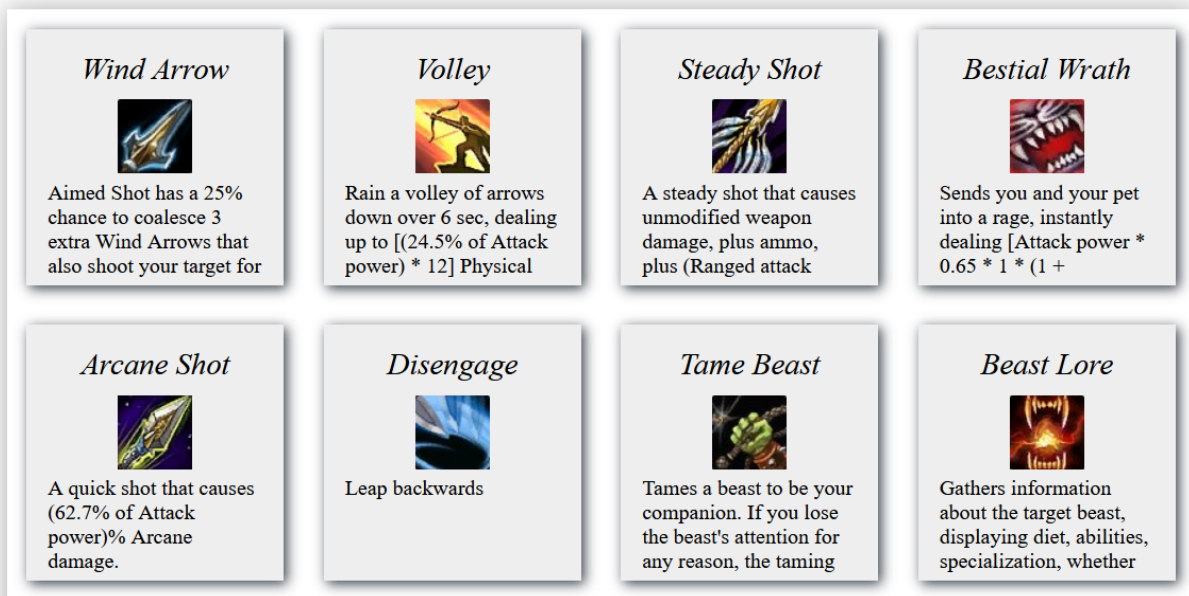
Zadanie składa się z implementacji serwera (Node.js) oraz klienta (witryna internetowa, React.js, Angular, html + js + css).

Część serwerowa:

- zaimplementuj serwer http w środowisku uruchomieniowym Node.js
- serwer http nasłuchuje na porcie `8080`
- serwer obsługuje jeden punkt dostępowy (endpoint), o adresie `egz_ai` oraz metodzie `GET`
- serwer powinien wysyłać wczytywany plik z dysku `data.json`
- opcjonalnie może wysyłać zawartość, która jest umieszczona w pamięci uruchomionej aplikacji
- pamiętaj o obsłudze błędu CORS (*Cross-Origin Resource Sharing*)

Część klientowa (witryna internetowa):

- po wczytaniu strony internetowej w przeglądarce należy wysłać żądanie do serwera w celu uzyskania danych
- na podstawie uzyskanych danych (można wykorzystać dane z pliku `data.json` w celu implementacji zadania), należy wygenerować poniższą stronę internetową:



Rysunek 1 Podgląd części strony internetowej

- główne ciało strony nie posiada żadnych marginesów wewnętrznych i zewnętrznych, jest koloru białego
- na środku strony, znajduje się kontener zawierające kafle pewnych akcji z gry komputerowej
- kontener ten posiada następujące atrybuty:
 - rozmiar: 895px
 - margines górny i dolny: 20px
 - margines prawy i lewy: automatyczny
 - cień (x,y) => 0px, (zaokrąglenie) => 20px, (kolor) => #ABABAB
- każdy kafel posiada następujące atrybuty:
 - kolor tła: #EEEEEE
 - ramkę: (kolor) => #EEEEEE, linia ciągła o rozmiarze 1px
 - cień: (x,y) => 2px, (zaokrąglenie) => 10px, (kolor) => #333C45

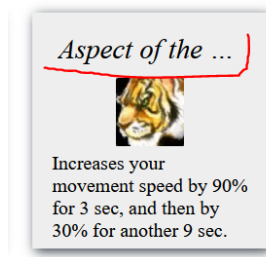
- rozmiar: (wysokość) => 192px, (szerokość) => 192px
- margines zewnętrzny: 15px

- po najechaniu myszki każdy kafel zmienia kolor ramki na czarny



- kafel składa się z dodatkowego kontenera, który posiada margines zewnętrzny 15px oraz następujących elementów: napis, obrazek oraz opis

- napis, jest wycentrowany, jego rozmiar to 23px, wykorzystana kursywa, margines zewnętrzny to 5px, atrybut ukrywający tekst, jeśli jest za długi i pojawiają się trzy kropki



- obrazek, jest wycentrowany, źródła obrazków znajdują się w images (serwer nie powinien ich udostępniać)

- opis, jego wysokość powinna być ograniczona do 75px a reszta tekstu powinna być ukryta

Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania. Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *internetowa* powinno znaleźć się archiwum całego projektu o nazwie *internetowa.zip*

Część II. Aplikacja okienkowa - notatnik

Napisać program, który odtworzy następujący wygląd okna:



Opis wyglądu aplikacji. Elementy od góry:

Program zawiera Menu z opcjami:

- Nowy
- Otwórz
- Zapisz
- Zapisz jako
- Zamknij

Program zawiera kontrolkę RichTextBox, która jest osadzona (zadokowana) w całym oknie programu. Gwarantuje to automatyczne dopasowanie kontrolki do zmieniającej się wielkości okna programu.

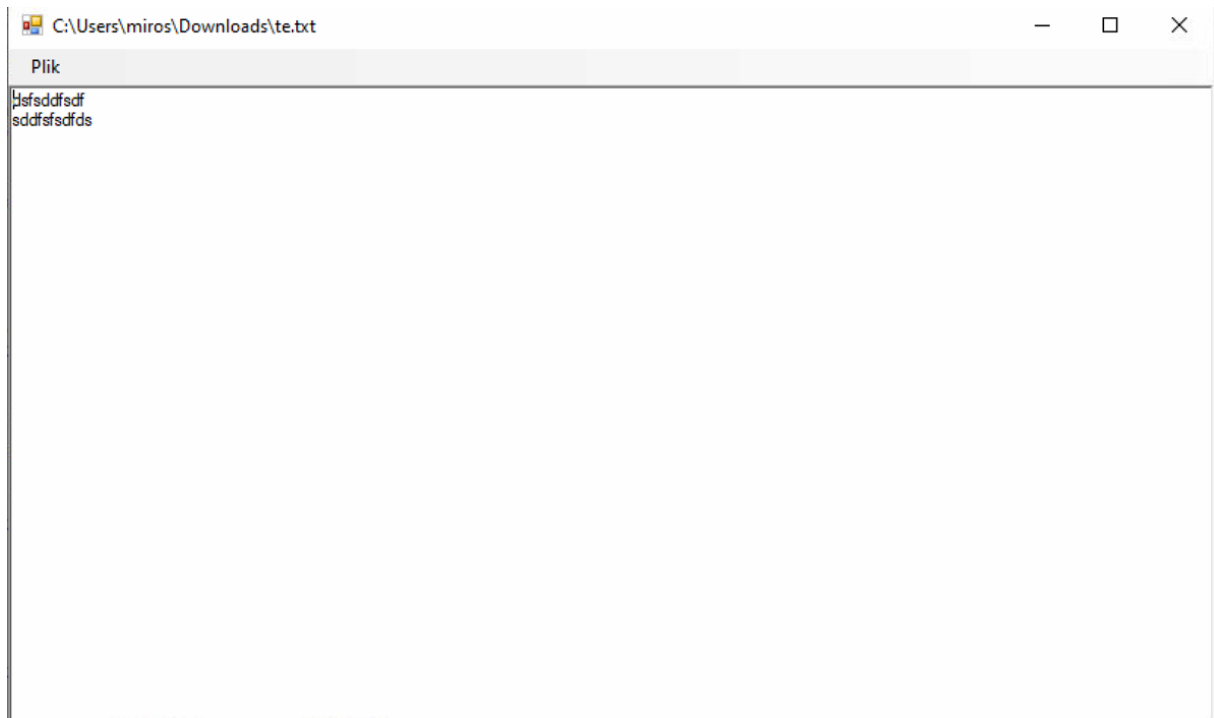
Program powinien na górnej belce okna wyświetlać nazwę: „Edytor tekstu” jeśli plik na którym pracujemy nie ma jeszcze określonej nazwy lub nazwę pliku na którym pracujemy.

Działanie aplikacji

1. Użytkownik może wprowadzać i edytować tekst w kontrolce RichTextBox.

2. Po wybraniu z menu opcji „Nowy” program czyści zawartość kontrolki RichTextBox i jeśli nazwa pliku jest wyświetlona na górnej belce okna, to zamiast nazwy wyświetlamy tekst „Edytor tekstu”

2. Po wybraniu opcji „Otwórz” pojawia się okno dialogowe OpenFileDialog w którym użytkownik wybiera plik, który następnie jest odczytywany i jego zawartość pojawia się w kontrolce RichTextBox. Nazwa odczytanego pliku pojawia się na górnej belce okna np.:



3. Po wybraniu z menu opcji „Zapisz”

- jeśli plik nie był zapisany pojawia się okno dialogowe SaveFileDialog w którym użytkownik wybiera miejsce zapisu pliku oraz podaje jego nazwę.

- jeśli plik już był zapisany lub odczytany to cała zawartość kontrolki jest zapisywana do pliku o wcześniej wybranej nazwie

4. Po wybraniu z menu opcji „Zapisz jako ...” pojawia się okno dialogowe SaveFileDialog w którym użytkownik wybiera miejsce zapisu pliku oraz podaje jego nazwę.

Po zapisaniu pliku jego nazwa powinna pojawić się na górnej belce okna.

Po wybraniu z menu opcji „Zamknij”, program powinien zakończyć swoje działanie.

W oknach OpenFileDialog oraz SaveFileDialog powinny być widoczne tylko pliki z rozszerzeniem txt

Założenia do programu

- Program wykonywany w oknie.
- Obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C# lub Java lub Python.

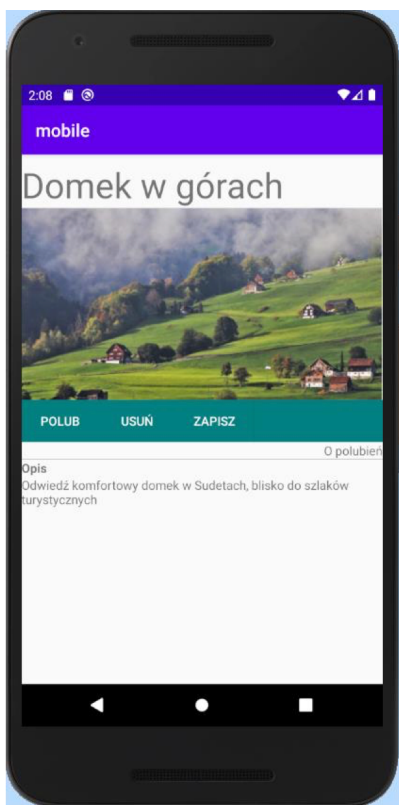
- Program powinien być zapisany czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych, funkcji i kontrolek.

Podjmij próbę kompilacji i uruchomienia. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *okienkowa* powinno znaleźć się archiwum całego projektu o nazwie *okienkowa.zip*

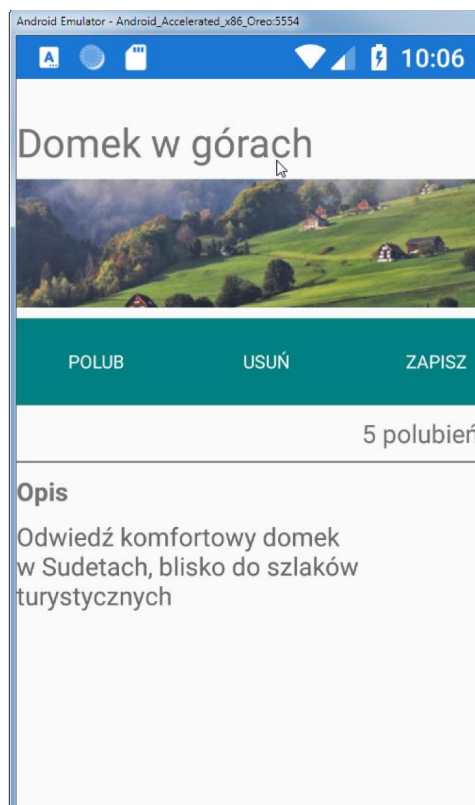
Część III. Aplikacja mobilna

Wykonaj aplikację mobilną za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Aplikacja jest fragmentem programu do przeglądania ofert turystycznych. Do wykonania aplikacji należy wykorzystać obraz z archiwum *zad1.zip*.



Obraz 1a.

Aplikacja AndroidStudio, stan początkowy. Emulacja Nexus 5X API 29 x86.



Obraz 1b.

Aplikacja MS Visual Studio, zachowanie aplikacji: 5 razy wciśnięto „POLUB”. Emulacja Android Oreo.

Na obrazie 1a przedstawiono stan po uruchomieniu aplikacji mobilnej. W zależności od zastosowanego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie się różnić od przedstawionego. Na obrazie 1b przedstawiono zachowanie aplikacji: 5 razy wciśnięto przycisk „POLUB” i w efekcie status pod przyciskami wyświetla tekst „5 polubień”.

Elementy aplikacji:

- Tytuł o treści: „Domek w górach”.
- Obraz o nazwie *obraz.jpg* wypakowany z archiwum.
- Trzy przyciski o treści: „POLUB”, „USUŃ”, „ZAPISZ” umiejscowione obok siebie.
- Napis o treści „0 polubień”.
- Linia horyzontalna.
- Napis o treści „Opis”.
- Napis o treści „Odwiedź komfortowy domek w Sudetach, blisko do szlaków turystycznych”.

Działanie aplikacji:

- Aplikacja implementuje licznik polubień, który w stanie początkowym aplikacji jest równy 0, następnie jego stan jest:
 - inkrementowany po wciśnięciu przycisku „POLUB”,
 - dekrementowany po wciśnięciu przycisku „USUŃ”. Licznik nie może być niższy niż 0.
- Stan licznika jest wyświetlany pod przyciskami, w formie napisu „<x> polubień”, gdzie <x> oznacza aktualną wartość licznika.

Założenia aplikacji:

- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML).
- Zastosowany typ rozkładu liniowy wertykalny (Linear / Stack lub inny o tej idei) z zagłębionym rozkładem liniowym horyzontalnym dla przycisków.
- Margines wewnętrzny górny dla całej strony lub rozkładu wertykalnego: 20 px (lub dp)
- Kolor tła przycisków i rozkładu, w którym się znajdują: Teal (#008080), zgodnie z Obrazem 1a.
- Kolory czcionki: biały dla przycisków oraz Gray (#808080) dla napisu „Odwiedź...”, zgodnie z Obrazem 1a.
- Czcionka tytułu ma rozmiar największy spośród użytych w aplikacji.
- Czcionka napisu „Opis” jest pogrubiona.
- Napis o liczbie polubień jest wyrównany do prawej.
- Obraz wypełnia całą szerokość strony (zależnie od zastosowanego aspektu może być automatycznie obcięty przez emulator – zobacz obraz 1b).
- Linia horyzontalna jest koloru Gray (#808080), dopuszcza się również prostokąt o wysokości 1.
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podjmij próbę kompilacji i emulacji. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *mobilna* powinno znaleźć się archiwum całego projektu o nazwie *mobilna.zip*

Część IV. Dokumentacja aplikacji

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej utwórz nagłówek funkcji przeszukującej, według wzoru zgodnie z listingiem 1. Nagłówek powinien znaleźć się w kodzie źródłowym nad funkcją. W miejscu znaków <> należy podać nazwę funkcji, nazwy argumentów wraz z krótkim opisem, typ zwracany wraz z krótkim opisem, krótki opis zawierający przynajmniej nazwę algorytmu. Gdy funkcja nie ma argumentów – zapisać „brak”. W miejscu autor należy podać swoje nazwisko i imię.

UWAGA: Dokumentację umieścić w komentarzu (wieloliniowym lub kilku jednoliniowych). Znajdujący się w listingu 1 wzór dokumentacji jest bez znaków początku i końca komentarza, gdyż te są różne dla różnych języków programowania

Listing 1. Wzór dokumentacji funkcji

```
*****
nazwa funkcji: <tu wstaw nazwę funkcji>
argumenty: <nazwa argumentu> - <co przechowuje>
           < nazwa argumentu> - <co przechowuje>
typ zwracany: <nazwa typu>, <co jest zwracane>
informacje: <opis>
autor: <Nazwisko i imię zdającego>
*****
```

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji utworzonych podczas egzaminu. Zrzuty powinny obejmować cały obszar ekranu monitora z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na rzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem lub okno terminala z kompilacją projektu. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na rzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wykonać należy tyle zrzutów ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja internetowa – dowolna liczba zrzutów nazwanych *internetowa1*, *internetowa2* ...
- Aplikacja okienkowa – dowolna liczba zrzutów nazwanych *window1*, *window2* ...
- Aplikacja mobilna – dowolna liczba zrzutów nazwanych *mobile1*, *mobile2* ...

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać podpisane zrzuty ekranu oraz zapisane informacje:

- o Nazwę systemu operacyjnego, na którym pracował zdający.
- o Nazwy środowisk programistycznych, z których zdający korzystał na egzaminie.
- o Nazwy języków programowania.
- o Nazwa emulatora systemu mobilnego.

Zrzuty ekranu i dokument umieścić w podfolderze *dokumentacja*.

Nagraj folder z rezultatami pracy. W folderze z Nazwiskiem i imieniem zdającego powinny się znajdować podfoldery: internetowa, okienkowa, mobilna, dokumentacja.

W folderze dokumentacja: pliki ze zrzutami oraz plik egzamin.

W folderze internetowa: spakowany cały projekt aplikacji internetowej, część serwerowa oraz klientowa

W folderze okienkowa: spakowany cały projekt aplikacji okienkowej, pliki z kodem źródłowym, opcjonalnie plik uruchomieniowy

W folderze mobilna: spakowany cały projekt aplikacji mobilnej, pliki ze źródłami interfejsu i logiki.

Upewnij się, że wszystko co stworzyłeś zostało prawidłowo nagrane