



**POLITECHNIK  
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,  
TELEKOMUNIKACJI I INFORMAT



Autorzy:

Mikołaj Klein

Piotr Kreft

Leszek Rudnicki

Forma studiów: niestacjonarne

Kierunek studiów: Informatyka

## **PROJEKT DYPLOMOWY**

Tytuł projektu:

**Aplikacja edukacyjna sterowana emocjami użytkownika**

data:

**30.08.2017**

|  |          |
|--|----------|
| <b>PRZEZNACZENIE I ZAŁOŻENIA APLIKACJI .....</b>                           | <b>3</b> |
| CEL EDUKACYJNY .....   | 3        |
| <i>Nauka materiału.....</i>  | 3        |
| <i>Stan emocjonalny ucznia.....</i>  | 3        |
| <i>Gra .....</i>   | 4        |
| UTRZYMYWANIE UCZNIA W OPTYMALNYM STANIE EMOCJONALNYM.....                  | 4        |
| <i>Serwis monitorujący stan emocjonalny.....</i>                           | 4        |
| <i>Techniki wpływania na stan ucznia .....</i>                             | 4        |
| CEL PRZEJŚCIA PRZEZ CAŁY MATERIAŁ .....                                    | 5        |
| <i>Wywołanie uczucia satysfakcji .....</i>                                 | 5        |
| <i>Utworzenie prostej, działającej gry.....</i>                            | 5        |
| <i>Możliwe gry.....</i>  | 5        |
| <b>ZASTOSOWANA METODA EDUKACJI .....</b>                                   | <b>5</b> |
| GRYWALIZACJA .....   | 5        |
| <i>Scenariusz.....</i>   | 5        |
| <i>Punktacja.....</i>  | 5        |
| <i>Tokeny .....</i>  | 6        |
| <i>Gry, losowania, itd.....</i>  | 6        |
| PROCES PRACY UCZNIA .....  | 6        |
| <i>Sposób podawania materiału.....</i>                                     | 6        |
| <i>Sterowanie przez ucznia.....</i>  | 6        |
| <i>Niezależna pętla monitorująca stan emocjonalny .....</i>                | 6        |
| <i>Stala obecność gry.....</i>   | 7        |
| STEROWANIE TRUDNOŚCIĄ MATERIAŁU .....                                      | 7        |
| <i>Przygotowane poziomy trudności materiału .....</i>                      | 7        |
| <i>Dobór poziomu trudności .....</i>                                       | 7        |
| <i>Strategia uczenia modułów, po których uczeń nie zaliczył testu.....</i> | 7        |
| DYSTRAKTORY .....  | 7        |
| <i>Typy dystraktorów.....</i>  | 7        |
| <i>Kontrola czasu ostatniego użycia.....</i>                               | 8        |
| <i>Losowanie.....</i>  | 8        |
| <b>KRYTYCZNE ELEMENTY APLIKACJI .....</b>                                  | <b>8</b> |
| POLA TABLICY 'MATERIAŁ' W BAZIE DANYCH .....                               | 8        |
| ZARZĄDZANIE MATERIAŁEM PRZEZ NAUCZYCIELA .....                             | 8        |
| TECHNOLOGIE .....  | 8        |
| <b>PROJEKT DZIAŁANIA APLIKACJI.....</b>                                    | <b>9</b> |
| GŁÓWNY ALGORYTM .....  | 9        |
| <i>Funkcja algorytmu .....</i>   | 9        |
| <i>Techniki, jakie algorytm ma do dyspozycji : .....</i>                   | 9        |
| REAKCJE NA 4 TYPY ZAPYTAŃ Z FRONTEND.....                                  | 9        |
| <i>"NASTĘPNY MODUŁ" .....</i>  | 9        |
| <i>"WYJAŚNIJ DOKŁADNIEJ".....</i>  | 10       |
| <i>"WYBRANY MODUŁ" .....</i>   | 10       |
| <i>"ZMIANA STANU POBUDZENIA".....</i>                                      | 10       |
| DIAGRAMY .....   | 11       |

# **Przeznaczenie i założenia aplikacji.**

## **Cel edukacyjny.**

### **Nauka materiału.**

Jest to aplikacja edukacyjna, której celem jest nauczenie użytkownika wybranego materiału.

Konstrukcja aplikacji umożliwia wypełnienie jej dowolnym materiałem. Warunkiem jest aby materiał można było podzielić na bardzo małe fragmenty wiedzy i utworzyć do nich testy. Test pojedynczego fragmentu materiału może być :

- testem wyboru,
- lub pytaniem otwartym.

Ponadto można zastosować specjalne zadania charakterystyczne dla danej dziedziny wiedzy. Niniejsza aplikacja uczy podstaw programowania. Teoria, przykłady i ćwiczenia przygotowane są dla języka C#. Aplikacja realizowana w niniejszym projekcie wykorzystuje kompilator kodu.

W przypadku innej dziedziny niezbędne byłoby utworzenie modułu dedykowanego do oceny charakterystycznych dla niej zadań. Np. dla nauki fizyki należałoby utworzyć moduł serwera oraz części frontend do tworzenia i weryfikacji wzorów fizycznych.

### **Stan emocjonalny ucznia.**

Aplikacja opiera się na założeniu, że nauka jest najbardziej efektywna, gdy uczący się jest w optymalnym stanie emocjonalnym - nie nudzi się ani nie frustruje.

#### **Co oznacza frustracja?**

- brak sukcesu przy motywacji do wykonania zadania - ułatwić
- konieczność uczenia się materiału, który ucznia nie interesuje - ułatwić, dystraktory, gra

#### **Co oznacza znudzenie?**

- materiał jest za łatwy - utrudnić
- materiał / dziedzina w ogóle ucznia nie interesuje - gra, dystraktory,

#### **Przyczyny nudy ucznia**

- zbyt rozwlekłe podawanie prostego materiału
- zbyt skomplikowane wyjaśnienie trudnym językiem
- zadania prawie się nie różnią - powtarza się ich idea
- brzydka szata graficzna programu

#### **Przyczyny frustracji ucznia**

- zbyt skomplikowany materiał (trzeba go podzielić na mniejsze)
- zbyt rozwlekłe podawanie materiału - uczeń już dawno wie o co chodzi (skoncentrować materiał)
- niemożliwe do przerywania / przyspieszenia elementy prezentacji (film, slajdy)

Powyższa analiza stanowi podstawę:

- algorytmu programu,
- podziału materiału, który będzie uczony na fragmenty,
- wyboru elementów gry i dystraktorów.

## **Gra**

Nauka wymaga motywacji. Zakładamy, że przeciętny uczeń często nie interesuje się dziedziną, której uczy aplikacja. Ma motywację szkolną - musi zaliczać przedmioty, ale jest to motywacja często słaba i negatywna (unikanie złej oceny). Taka motywacja nie sprzyja skutecznej nauce.

Znacznie lepsza jest motywacja pozytywna. Daje lepsze wyniki nauki oraz lepszą jakość życia uczniowi - zamiast przechodzić obowiązkowe tortury taki uczeń dąży do własnego, uwewnętrznionego celu, który przyniesie mu nagrodę.

Dlatego aplikacja wykorzystuje grwalizację. W tym celu przygotowana jest do nałożenia na "suchy" materiał gry:

- materiał umieszczony jest w prostym scenariuszu nadającym koloryt pracy ucznia,
- wszystkie zadania przynoszą uczniowi punkty w grze,
- w grze nigdy się nie traci - można najwyżej nie wygrać,
- uczeń na bieżąco widzi swój wynik, otrzymuje nagrody specjalne i zadania nie związane z materiałem.

## **Utrzymywanie ucznia w optymalnym stanie emocjonalnym.**

### **Serwis monitorujący stan emocjonalny**

Dla zwiększenia efektywności uczenia się aplikacja wykorzystuje webservice, który wykrywa 3 stany emocjonalne na podstawie obrazu twarzy:

- znudzenie
- frustrację
- brak znudzenia i frustracji

Oprócz podawania i testowania materiału aplikacja stale monitoruje stan emocjonalny ucznia. Wykrycie stanu nieoptymalnego (znudzenia lub frustracji) powoduje reakcję aplikacji, mającą na celu przywrócenie ucznia do stanu optymalnego.

### **Techniki wpływania na stan ucznia**

Aplikacja podaje kolejne partie materiału w reakcji na żądanie ze strony ucznia. Równocześnie reaguje na:

- stan emocjonalny ucznia,
- jego dotychczasowe wyniki.

W przypadku wykrycia nieoptymalnego stanu emocjonalnego ucznia aplikacja reaguje próbując przywrócić go do stanu optymalnego. W tym celu dobiera:

- stopień trudności materiału (łatwiejszy, gdy uczeń się frustruje, trudniejszy, gdy nudzi),
- dystraktory stanowiące przerywnik w pracy, pobudzające, rozluźniające,
- elementy gry aby budować motywację do przezwyciężenia trudności oraz wzbudzić w uczniu uczucia sukcesu, które sprzyja kontynuacji wysiłku.

Założono, że kontrola nad wyświetlanym materiałem korzystnie wpływa na zaangażowanie ucznia oraz redukuje frustrację i nudę. Dlatego uczeń otrzymuje kontrolę nad tym jaki materiał otrzyma:

- widzi listę wszystkich modułów materiału i może w każdej chwili do nich wracać lub otwierać dalsze,
- ma do dyspozycji funkcję "wyjaśnij dokładniej", która podaje materiał w prostszej, dokładniej wyjaśnionej wersji,
- niektóre elementy gry uczniowi wybór co chce robić dalej aby uzyskać maksimum dodatkowych punktów.

## **Cel przejścia przez cały materiał**

### **Wywołanie uczucia satysfakcji**

Przyjęto, że dla utrwalenia wyników nauki oraz nadania dodatkowego sensu wykonywanym ćwiczeniom korzystne będzie, jeśli wykonanie wszystkich ćwiczeń zaowocuje utworzeniem jakiejś działającej całości. W ten sposób uczeń odczuje dodatkową satysfakcję z wykonanego dzieła, a ta poprawi jego nastawienie do dziedziny, której się uczył. Dzięki temu zaistnieje szansa, że będzie się w niej rozwijał dalej, a jeśli nie jest ona przedmiotem jego zainteresowań to przynajmniej łatwiej mu będzie opanować materiał tego przedmiotu wymagany przez szkołę.

### **Utworzenie prostej, działającej gry**

W tym celu materiał do ćwiczeń przygotowano w następujący sposób:

- uczeń w trakcie ćwiczeń pisze ćwiczenia,
- każde ćwiczenie stanowi fragment końcowego kodu,
- końcowy kod to bardzo prosta, działająca gra,
- najlepiej jeśli cały kod będzie napisany przez ucznia, ale można trudniejsze fragmenty, ew. grafikę itd. - dostarczyć z zewnątrz i umieścić kod ucznia w odpowiednich miejscach.

W tę grę uczeń będzie mógł zagrać i zobaczyć wyniki swojej pracy.

### **Możliwe gry**

Przykładowe gry, które można zrealizować w najprostszej wersji na planszy, w trybie znakowym:

- labirynt - budowa labiryntu, przejście przez labirynt
- czołgi - plansza z przeszkodą (ściana), losowe ruchy przeciwników
- space invaders - losowe ruchy przeciwników (zestrzelić), narastająca liczba przeciwników
- pong
- bullet hell w jednym wymiarze - spadają pociski (uciec) i wrogowie (zestrzelić)

## **Zastosowana metoda edukacji.**

### **Grywalizacja**

#### **Scenariusz**

Przed rozpoczęciem użytkownik wybiera w jakim chce być scenariuszu - np.:

- sport - konkurujesz z innymi sportowcami / drużyna z drużynami
- uczysz się na hackera
- Matrix - Neo, Morfeusz, Wyrocznia, Smith, mątwy, itd.
- Władca Pierścieni lub inny własny scenariusz fantasy
- scenariusze dla dziewczyn..?
- 

#### **Punktacja**

Każde test i ćwiczenie stanowi zadanie lub misję w grze. Zależnie od stopnia trudności oraz aktualnego poziomu ucznia (etap nauki) przynosi określoną liczbę punktów.

Punktacja uzyskana do tej pory jest stale widoczna w GUI użytkownika.

## Tokeny

Oprócz punktów uczeń zdobywa również inne nagrody w grze:

- odznaki,
- tytuły,
- tokeny, które można wymieniać na coś wartościowego dla ucznia (np. przedłużenie czasu testu, który ma ograniczenie czasowe, dodatkową sesję jednej z małych gier używanych przez aplikację jako dystraktory, itd).

Odznaki, tytuły i tokeny są stale widoczne w GUI użytkownika, podobnie jak jest to stosowane w zwykłych grach komputerowych.

## Gry, losowania, itd

Elementem gry są również:

- małe, krótkie, minutowe gry - uczeń co jakiś czas otrzyma możliwość przeznaczenia kilku minut na taki przerywnik,
- koło fortuny, w którym uczeń może wygrać punkty.

Ich funkcje to:

- dystraktory, do wykorzystania przez główny algorytm - mogą zarówno pobudzić znudzonego jak rozluźnić sfrustrowanego,
- wzmocnienie klimatu gry o punkty, który daje motywację do rozwiązywania zadań.

## Proces pracy ucznia

### Sposób podawania materiału

Każdy fragment materiału zawiera

- materiał teoretyczny
- przykład
- test lub / oraz zadanie

Uczeń zdobywa punkty rozwiązując testy i zadania.

### Sterowanie przez ucznia

Moduły materiału podawane są w wyniku wciśnięcia przez ucznia przycisków

- 'następny' - podaje kolejny moduł materiału na ścieżce narastającej trudności i budowania całościowej wiedzy
- 'wybrany' - podaje moduł, który uczeń wybrał z listy wszystkich modułów,
- 'wyjaśnij' - jeżeli aktualnie podany moduł nie jest najprostszą postacią materiału - podaje prostszą postać

Obsługując żądanie wywołanego przez każdy z tych przycisków, lub menu serwer sprawdza najnowszy stan emocjonalny ucznia.

### Niezależna pętla monitorująca stan emocjonalny

Niezależnie od aktywności ucznia aplikacja cyklicznie sprawdza stan emocjonalny ucznia. W przypadku zmiany tego stanu wysyła żądanie reakcji do serwera.

W takim przypadku, jeżeli stan z optymalnego zmienił się na niepożądany, serwer wysyła do ucznia propozycję. Propozycja jest zależna od sytuacji:

- przesłanie tego samego materiału w łatwiejszej bądź trudniejszej wersji,
- dystraktor odpowiedni do stanu ucznia.

Uczeń może ale nie musi skorzystać z propozycji. Nadrzędnym kryterium jest kontrola ucznia nad materiałem.

## Stała obecność gry

W GUI stale widoczny jest stan gry użytkownika. Kolejne zadania 'opakowane' są w klimat gry - np. zadania mogą nosić nazwę 'misji' i mieć krótki fragment scenariusza umiejscawiający je w całym scenariuszu gry. Np. test wyboru w scenariuszu 'Matrix' mógłby zostać wyświetlony jako: 'Twoja misja: wszedłeś w Matrix i musisz otworzyć drzwi w korytarzu Architekta podając odpowiedź na to pytanie: *<tu wstawić test wyboru>* '.

Każde wykonane zadanie podnosi punktację i przynosi korzyści w grze. Np. token zdobyty wcześniej może w powyższym przykładzie służyć do uzyskania dodatkowego podejścia do zadania lub dłuższego czasu na rozwiązanie, itd.

## Sterowanie trudnością materiału

### Przygotowane poziomy trudności materiału

Materiał przygotowany jest w 3 wersjach różniących się poziomem trudności

- podstawowy : moduły bardzo małe, do każdego elementu przykład
- łączony: połączenie 3 do 6 modułów podstawowych do wyświetlenia razem; następnie test również jest połączeniem testów do każdego z modułów podstawowych
- skoncentrowany : moduły duże, napisane bardziej lakonicznie, mniej przykładów lub przykład łączący kilka elementów materiału, trudniejsze testy.

### Dobór poziomu trudności

Główny algorytm programu wybiera poziom trudności kolejnego modułu, reagując na stan emocjonalny gracza i jego dotychczasowe wyniki. Jest on opisany w dalszej części tej dokumentacji.

### Strategia uczenia modułów, po których uczeń nie zaliczył testu

Pula reakcji do wyboru:

- powtórzyć moduł od razu w łatwiejszej wersji,
- wstawić moduł w łatwiejszej wersji do kolejki o kilka miejsc dalej - zostanie wywołany ponownie,
- zaproponować trochę później zadanie specjalne w grze - 3 najtwardsze orzechy do zgryzienia (3 nie zaliczone moduły) - i test dający podwójne punkty, specjalny token, itp.,
- specjalnie wyróżnić graficznie / kolorem na liście modułów te, których nie zaliczył, żeby mógł łatwo do nich wrócić (nie używając koloru czerwonego, które niesie kulturowe konotacje takich jak "przegrana", "zatrzymanie", "problem" - zamiast tego używając konotacji: "zadanie", "tu jest jeszcze coś do zrobienia", "pole minowe", itd. i oznaczając na zielono moduły zaliczone).

## Dystraktory

### Typy dystraktorów

Nagroda w grze (dla sfrustrowanego lub znudzonego) :

- gra w sapera - dodaje punkty do grywalizacji, im lepszy wynik tym więcej (mam gotową w javascript)
- koło fortuny - można wygrać punkty, tokeny, tytuły

Pobudzający (tylko dla znudzonego) :

- coś przebiega przez ekran w ciągu sekundy i znika
- ekran 'Matrix'
- żart rysunkowy
- dowcip (o programistach, oczywiście)

## Kontrola czasu ostatniego użycia

Aplikacja ma uniknąć powtarzalności i znudzenia dystraktorami. Dlatego moduł zarządzania dystraktorami:

- pobiera tylko dystraktory o wymaganym charakterze,
- sprawdza czas ostatniego użycia każdego z nich,
- wybiera tylko spośród użytych dawniej,
- rejestruje czas użycia dystraktora.

## Losowanie

Jeżeli w danym momencie do kryteriów pasuje kilka dystraktorów - wybór jednego z nich jest losowany.

# Krytyczne elementy aplikacji.

## Pola tablicy 'materiał' w bazie danych

Te pola odzwierciedlają sposób użycia jednostki materiału.

Pola:

- id (m.in. pozwala powiązać scenariusz gamifikacji z tym modulem)
- id grupy (dla utworzenia jedn. łączonej)
- typ: łatwa, trudna (szczegółowość wyjaśnienia materiału)
- temat (pokaże się na liście przerobionych modułów)
- materiał (treść do nauczania)
- przykład (przykład użycia tego materiału)
- typ testu (test wyboru, pytanie otwarte, zadanie z kodu)
- treść testu
- odpowiedź do testu
- punktacja w grywalizacji

## Zarządzanie materiałem przez nauczyciela

Nauczyciel to drugi typ użytkownika - ma uprawnienie do edycji materiału.

Nauczyciel ma dodatkową funkcjonalność w GUI frontend - prawy klik rozwija edycję danego elementu materiału.

Poza tym otrzymuje ten sam frontend co uczeń - to pozwala mu obejrzeć wynik edycji materiału tak, jak go zobaczy uczeń.

## Technologie

komunikacja: JSON, ASP NET WEB API

frontend: Angular 2, TypeScript

serwer: IIS, C#

baza danych: Postgres



# **Projekt działania aplikacji.**

## **Główny algorytm**

### **Funkcja algorytmu**

Główny algorytm aplikacji podejmuje decyzję o tym jaki materiał wysłać do użytkownika.

- reaguje na 4 rodzaje zapytania:
  - o następny moduł (przycisk)
  - o wyjaśnij dokładniej (przycisk)
  - o poprzedni moduł (przycisk)
  - o zmiana stanu pobudzenia (śledzona w pętli, niezależnie od działań ucznia)
- w kolejnym module materiału wysyła
  - o materiał lub test (wg. ścieżki podawania materiału)
  - o moduł scenariusza grywalizacji
  - o aktualizuje listę przerobionych modułów
- w reakcji na wynik testu wysyła
  - o wynik tego testu
  - o jeżeli to sukces:
    - aktualizację łącznego wyniku uzyskanego do tej pory
    - elementy grywalizacji (nagrody dodatkowe, uzyskany status, token, itd.)
  - o jeżeli wynik błędny i to było ćwiczenie z kodu - informacje z kompilatora pomagające w usunięciu błędu

### **Techniki, jakie algorytm ma do dyspozycji :**

- ułatwienie / utrudnienie materiału
- dystraktory
- elementy grywalizacji nie związane z materiałem a dające punkty - zakładamy, że nagroda, czyli punkty, pomagają zarówno pobudzić przy znudzeniu jak uspokoić frustrację dzięki poczuciu sukcesu

## **Reakcje na 4 typy zapytań z frontend**

### **"NASTĘPNY MODUŁ"**

Wywoływany jest przyciskiem 'dalej' w UI:

- stałe widocznym
- ale tylko jeżeli to nie jest ostatni moduł materiału

Przy każdej decyzji algorytm sprawdza:

- czy wszystkie wcześniejsze moduły niezbędne dla kolejnego zostały zaliczone
- stan emocjonalny (otrzymany z frontend)
- dotychczasowe wyniki testów z materiału (dobre / złe - z bazy)
- aktualny poziom trudności materiału
- aktualny moduł materiału
- jak długo uczeń znajduje się w nieoptymalnym stanie emocjonalnym

**ZNUDZENIE** - pula reakcji do wyboru:

- zależnie od wyników
  - o wyniki wysokie:
    - i. dystraktor pobudzający
    - ii. utrudnić materiał

- wyniki niskie:
  - iii. dystraktor: nagroda w grze
  - iv. lub dystraktor rozśmieszający, pobudzający, itp.
- zaproponować uczniowi wybór z puli różnorodnych zadań dających punkty

FRUSTRACJA - pula reakcji do wyboru:

- ułatwić materiał
- dać łatwe zadanie specjalne, za które uczeń otrzyma dodatkowe punkty (będzie miał poczucie sukcesu - dobrać zadanie łatwe w stosunku do aktualnego poziomu trudności)
- zaproponować wybór z puli zadań (w której zadania będą łatwe)
- dystraktor rozśmieszający, rozluźniający, itp

STAN OPTYMALNY

- wysłać kolejny moduł na tym samym poziomie trudności

## "WYJAŚNIJ DOKŁADNIEJ"

Wywoływany przyciskiem 'wyjasnij' w UI:

- stale widocznym
- ale tylko gdy to nie jest najłatwiejsza forma tego materiału

Akcja:

- wysłać ten sam materiał w postaci łatwiejszej
- po kliknięciu 'następny' wrócić do poziomu trudności z którego kliknięto 'wyjaśnij'

## "WYBRANY MODUŁ"

Wywoływany wciśnięciem pozycji na liście wszystkich modułów.

Akcja:

- wysłać wybrany moduł na poziomie, na którym który wysłano go wcześniej
- dać wybór uczniowi czy chce zrobić test (test daje możliwość zaliczenia niezaliczonego modułu, a przez to podniesienia swoich punktów)
- przycisk 'Wróć' przerzuca z powrotem do modułu, z którego kliknięto otwieranie modułu wcześniejszego

## "ZMIANA STANU EMOCJONALNEGO"

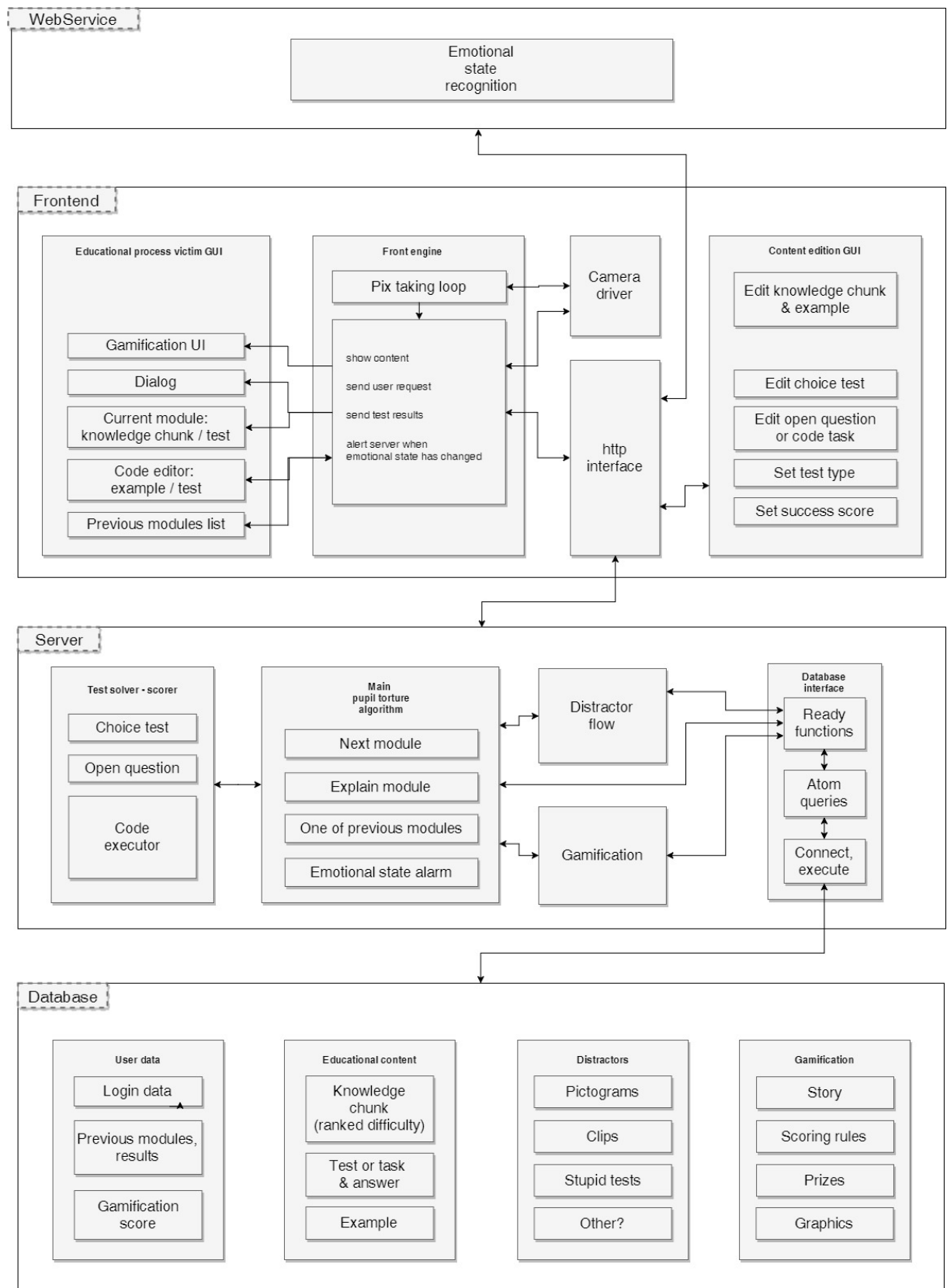
Wywoływana automatycznie przez frontend po wykryciu zmiany stanu przez pętlę.

Akcja:

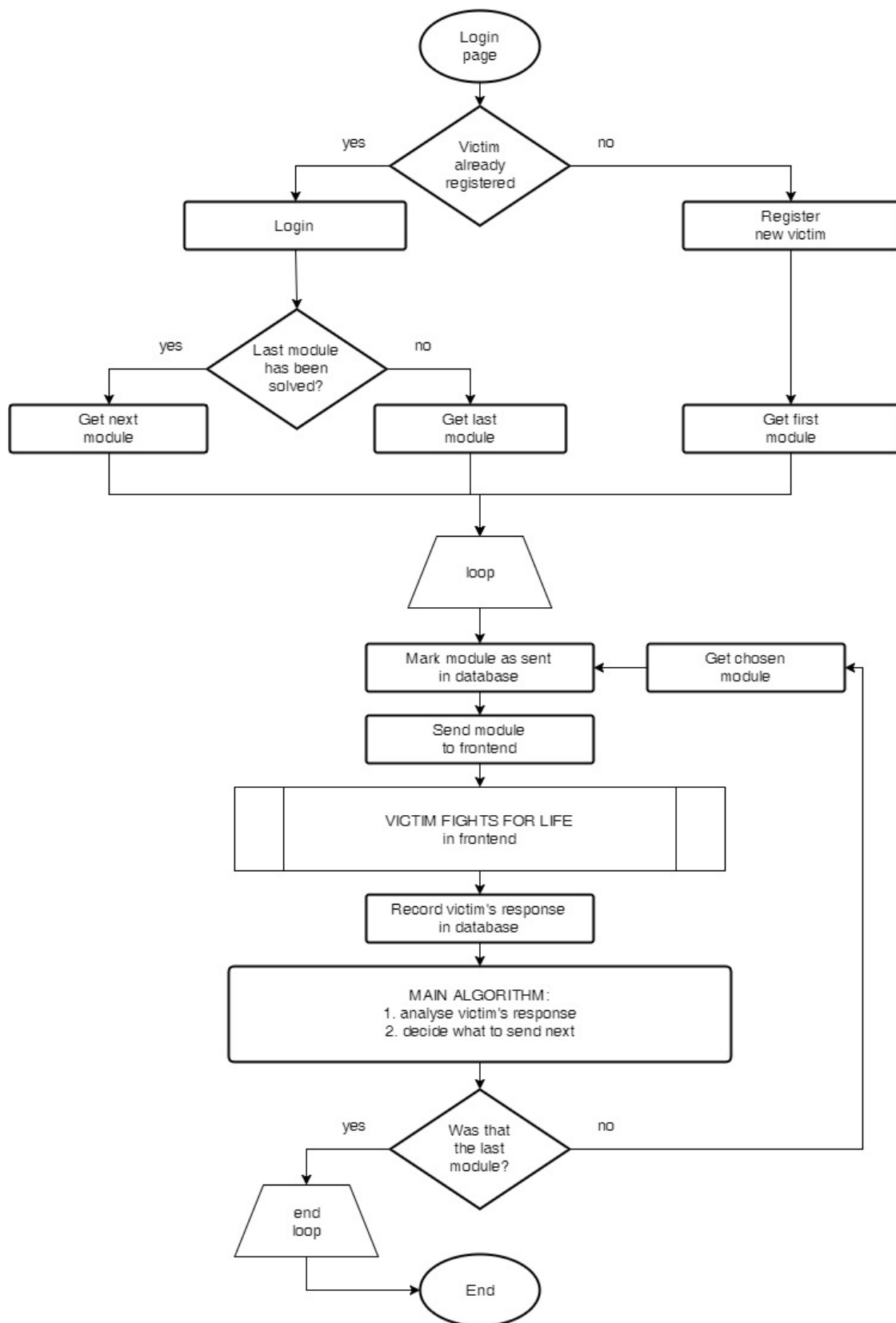
- jeśli to jest wejście we frustrację:
  - zaproponować ten sam materiał w postaci łatwiejszej
  - zaproponować łatwiejsze zadanie
  - dystraktor z nagrodą
- jeśli to jest wejście w znudzenie
  - dystraktor pobudzający
  - inne ?
- przesłać proponowany materiał tylko, jeśli uczeń zaakceptuje przyciskiem propozycję
- (zarejestrować metodę / zdanie, które wywołała ten stan - dla celów parametryzacji i zwiększania efektywności programu)

# Diagramy

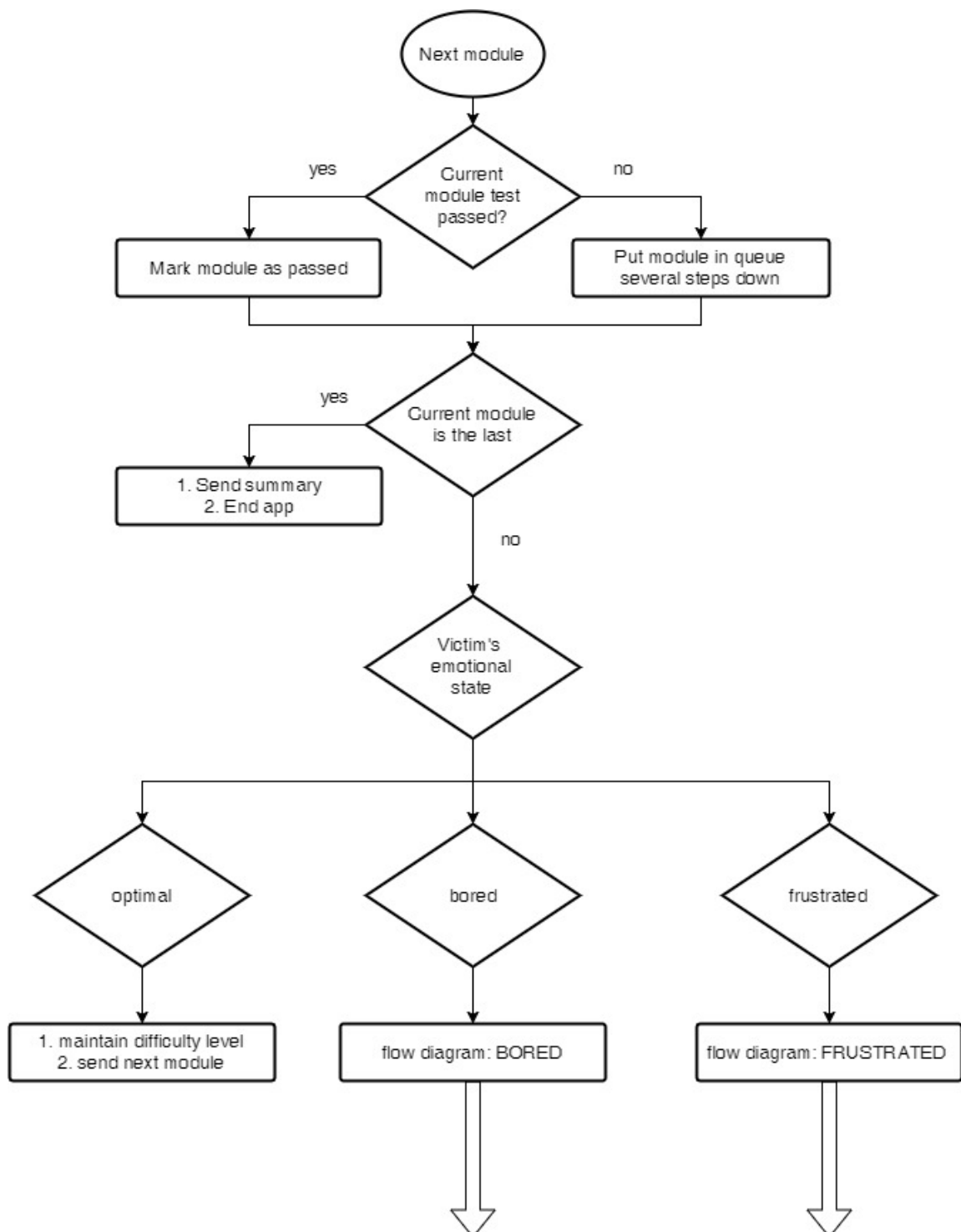
## Architektura - główne moduły



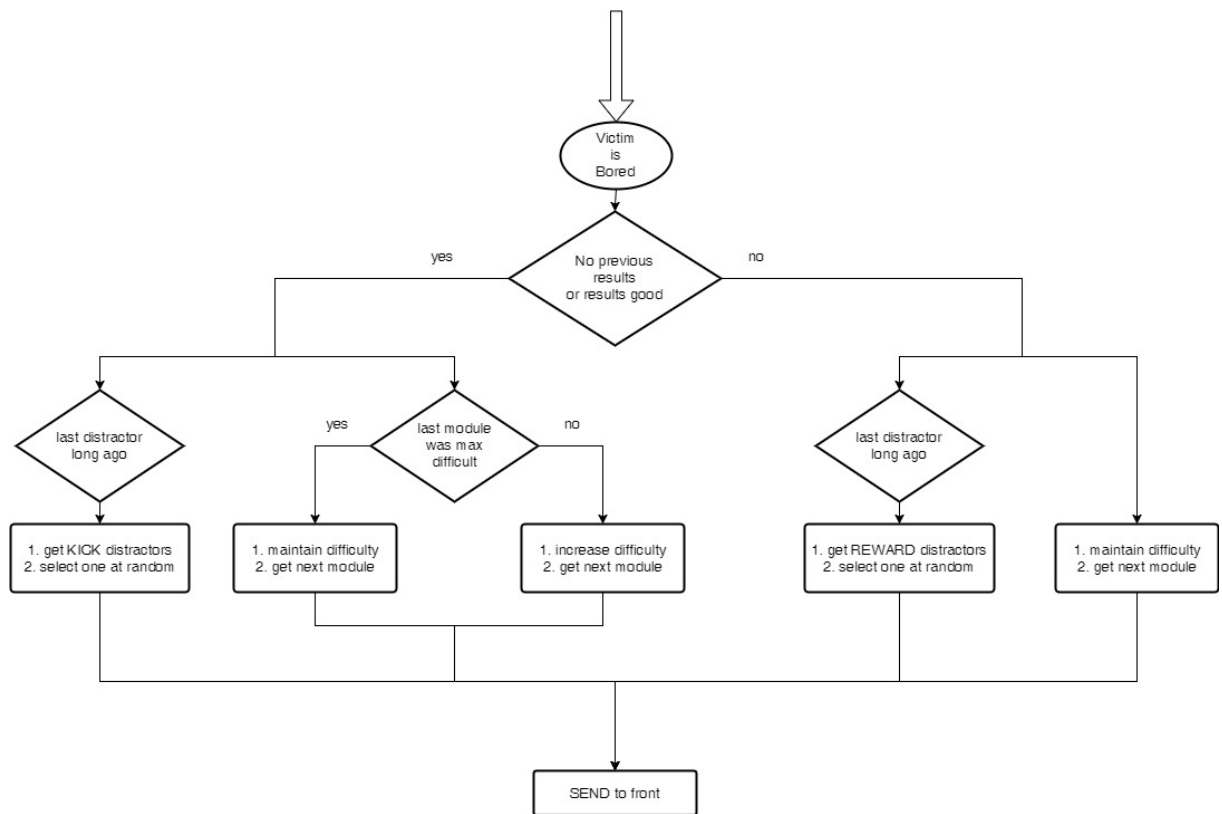
## Główna pętla programu



## Główny algorytm



## Kolejny moduł, gdy uczeń jest znudzony



## Kolejny moduł, gdy uczeń jest sfrustrowany

