Politechnika Krakowska im. Tadeusza Kościuszki

Wydział Mechaniczny

Dokumentacja

Programowanie aplikacji dla urządzeń mobilnych

Temat: Edukacyjna aplikacja dla dzieci i młodzieży - rozwiązywanie prostych równań matematycznych, umiejętność skutecznego zapamiętywania oraz rozwiązywanie testu IQ.

Autor: Paweł Wójcik
Numer grupy: 42K9
Numer zespołu: 13

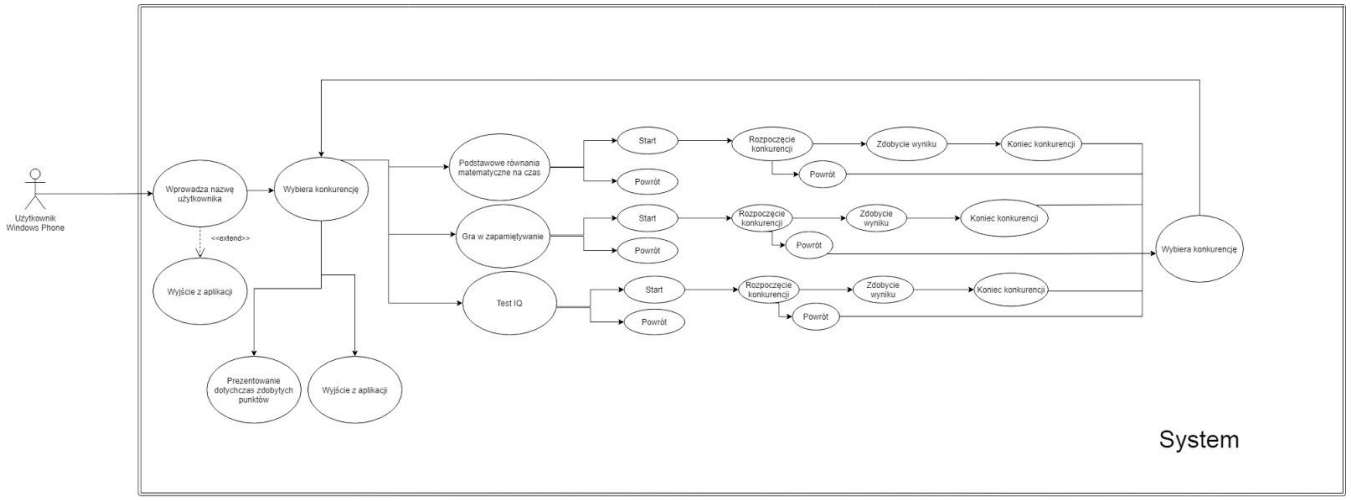
1.Wymagania funkcjonalne:

- rozdzielczość dopasowana do rozdzielczości ekranu telefonu
- aplikacja wyposażona w menu
- sterowanie odbywa się za pomocą wyświetlacza
- aplikacja posiada trzy konkurencje
- moduł podstawy równań matematycznych
- moduł gra w zapamiętywanie par
- moduł test IQ
- zdobywanie punktów i podsumowywanie
- przyjazna kolorystyka dla użytkownika

2.Wymagania niefunkcjonalne:

- aplikacja działa na platformie Windows Phone 8.1
- aplikacja wyposażona w intuicyjny i przejrzysty interfejs użytkownika
- atrakcyjna prezentacja wyników
- agregacja najważniejszych informacji
- zrozumiałość - aplikacja ma być prosta w użytkowaniu
- implementacja - C#, xaml, Windows Phone 8.1

3.Diagram przypadków użycia i scenariusze:



Scenariusze Przypadków Użycia

Id: UC1
Nazwa: Nazwa użytkownika - nickname - domyślny
Aktorzy: Użytkownik Windows Phone
Cel: Nie wprowadzanie nazwy użytkownika
Scenariusz:

1. Zaznaczenie pola tekstowego
2. Pozostawienie pustego pola tekstowego
3. Zatwierdzenie przyciskiem “Next”
4. Przydzielenie domyślnej nazwy użytkownika o nazwie “default”

Id: UC2
Nazwa: Nazwa użytkownika - nickname
Aktorzy: Użytkownik Windows Phone
Cel: Wprowadzenie nazwy użytkownika
Scenariusz:

1. Zaznaczenie pola tekstowego
2. Wprowadzenie nazwy użytkownika
3. Zatwierdzenie przyciskiem “Next”

Id: UC3
Nazwa: Konkurencja - podstawowe równania matematyczne
Aktorzy: Użytkownik Windows Phone
Cel: Wykonanie konkurencji
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie pierwszej konkurencji - podstawowe równania matematyczne
3. Zatwierdzenie przyciskiem “Start”

Id: UC4
Nazwa: Pośrednie wyniki - 1
Aktorzy: Użytkownik Windows Phone
Cel: Wyświetlenie pośrednich wyników - podstawowe równania matematyczne
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie pierwszej konkurencji - podstawowe równania matematyczne
3. Zatwierdzenie przyciskiem “Start”
4. Rozwiązanie określonej ilości równań
5. Wyświetlenie pośrednich wyników z bieżącej konkurencji

Id: UC5
Nazwa: Konkurencja - gra w zapamiętywanie
Aktorzy: Użytkownik Windows Phone
Cel: Wykonanie konkurencji
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie drugiej konkurencji - gra w zapamiętywanie
3. Zatwierdzenie przyciskiem “Start”

Id: UC6
Nazwa: Pośrednie wyniki - 2
Aktorzy: Użytkownik Windows Phone
Cel: Wyświetlenie pośrednich wyników - gra w zapamiętywanie
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie drugiej konkurencji - gra w zapamiętywanie
3. Zatwierdzenie przyciskiem “Start”
4. Dobranie wszystkich zadanych par lub koniec czasu
5. Wyświetlenie pośrednich wyników z bieżącej konkurencji

Id: UC7
Nazwa: Konkurencja - test IQ
Aktorzy: Użytkownik Windows Phone
Cel: Wykonanie konkurencji
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie trzeciej konkurencji - gra w zapamiętywanie
3. Zatwierdzenie przyciskiem “Start”

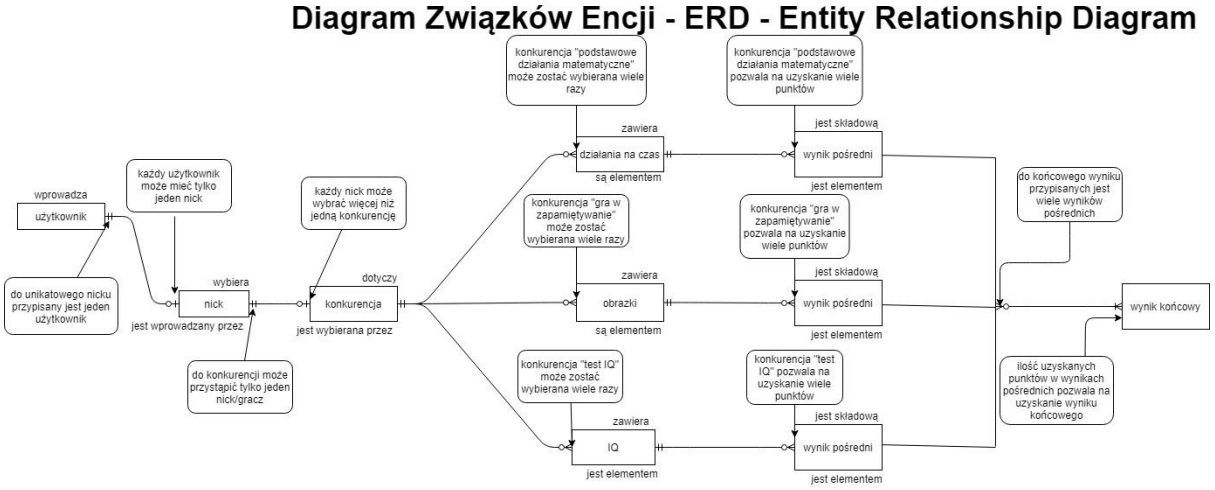
Id: UC8
Nazwa: Pośrednie wyniki - 3
Aktorzy: Użytkownik Windows Phone
Cel: Wyświetlenie pośrednich wyników - test IQ
Scenariusz:

1. Wpisanie nazwy użytkownika
2. Wybranie trzeciej konkurencji - gra w zapamiętywanie
3. Zatwierdzenie przyciskiem “Start”
4. Rozwiązanie określonej ilości zadań lub koniec czasu
5. Wyświetlenie pośrednich wyników z bieżącej konkurencji

Id: UC9
Nazwa: Wyniki
Aktorzy: Użytkownik Windows Phone
Cel: Uzyskanie wyników
Scenariusz:

1. Przejście do podsumowania wyników
2. Przedstawienie wyników w postaci liczb

4. Diagram bazy danych Związków Encji - ERD - **E**ntity-**R**elationship **D**iagram:

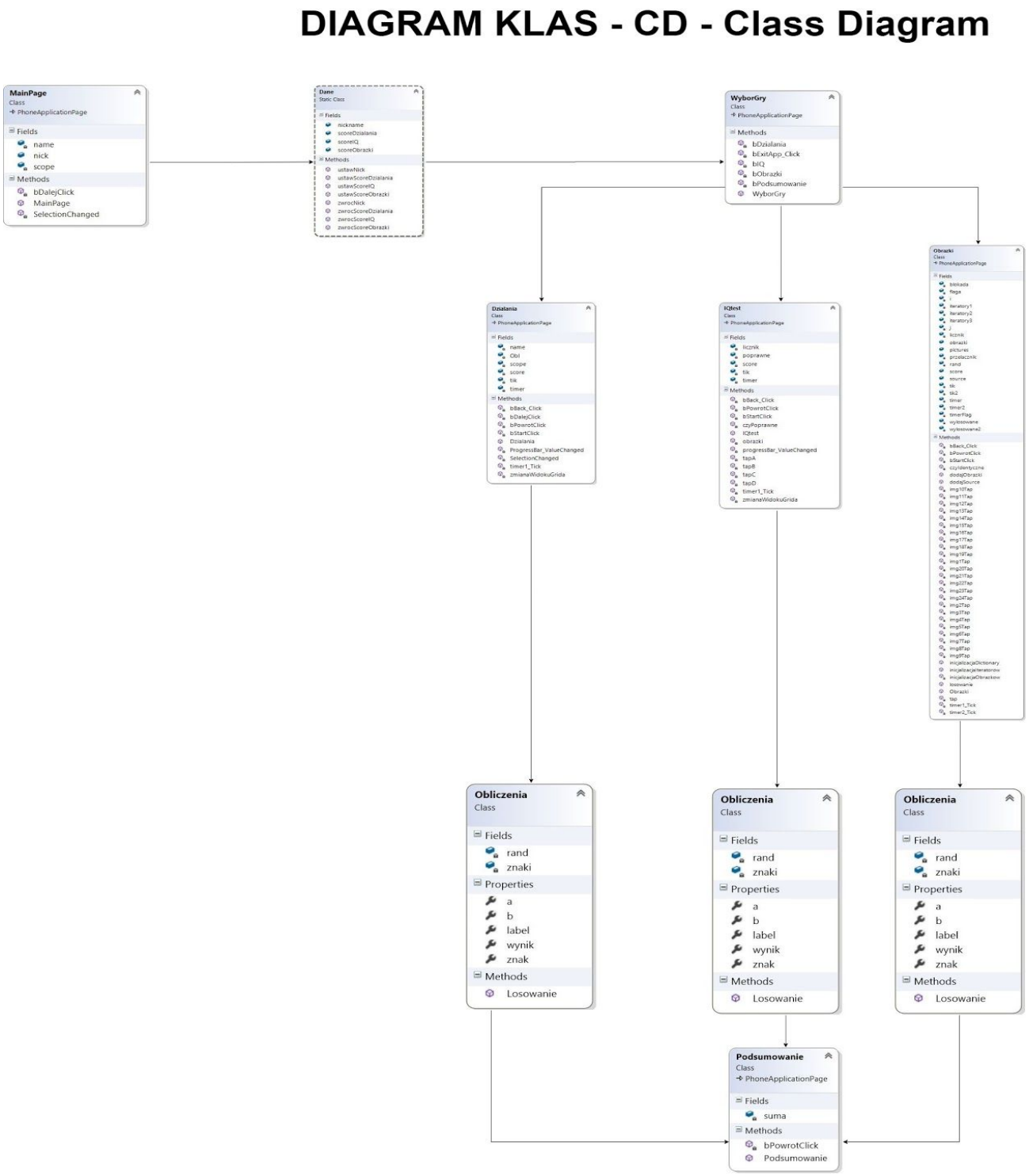


5. Lider zespołu: Paweł Wójcik

6. Wstępny podział zadań: Paweł Wójcik
   - Założenia wstępne
   - Pierwsze działające moduły aplikacji
   - Finalna wersja aplikacji
   - Moduł podstawy równań matematycznych
   - Moduł gra w zapamiętywanie par
   - Moduł test IQ
   - Zdobywanie punktów i podsumowywanie
   - Interfejs użytkownika
   - Wymagania funkcjonalne
   - Wymagania niefunkcjonalne
   - Diagram przypadków użycia oraz scenariusze - UCD - Use Case Diagram
   - Diagram związków encji ERD - Entity Relationship Diagram
   - Diagram klas CD - Class Diagram
   - Diagram przepływu danych - DFD - Data Flow Diagram
   - Diagram stanów - STD - State Diagram
   - Zrzuty ekranu

7.System kontroli wersji:
Projekt jest utworzony w systemie kontroli wersji git, dostępny na serwerze GitHub pod następującym adresem:
https://github.com/wojcikpawel/EducationalGameRepository

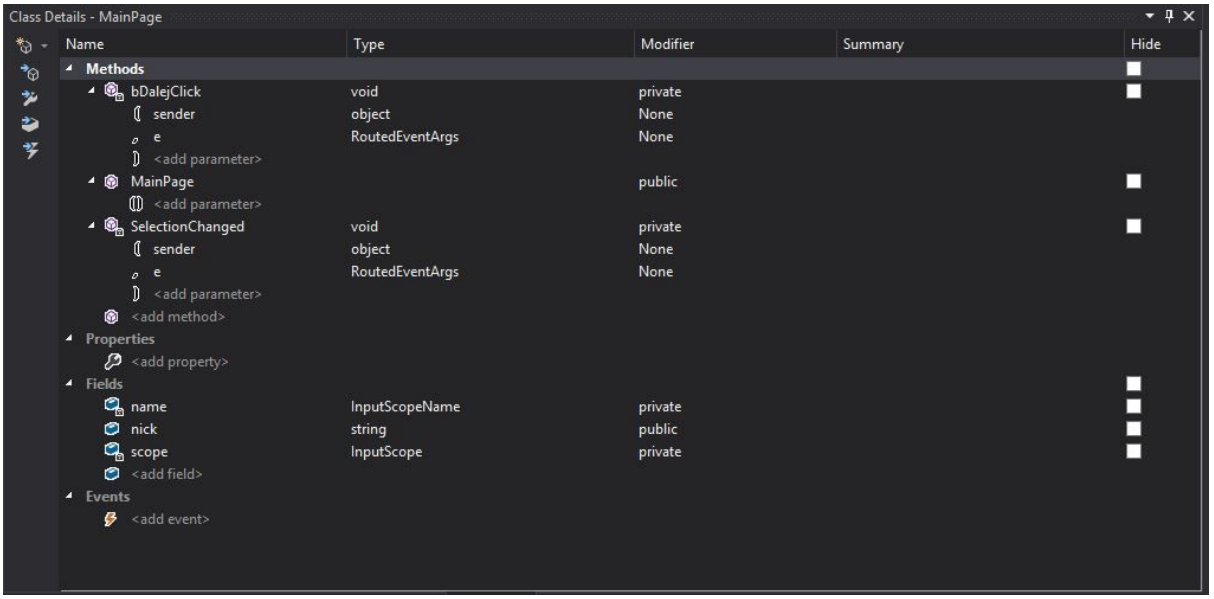
8. Diagram klas - CD - Class Diagram



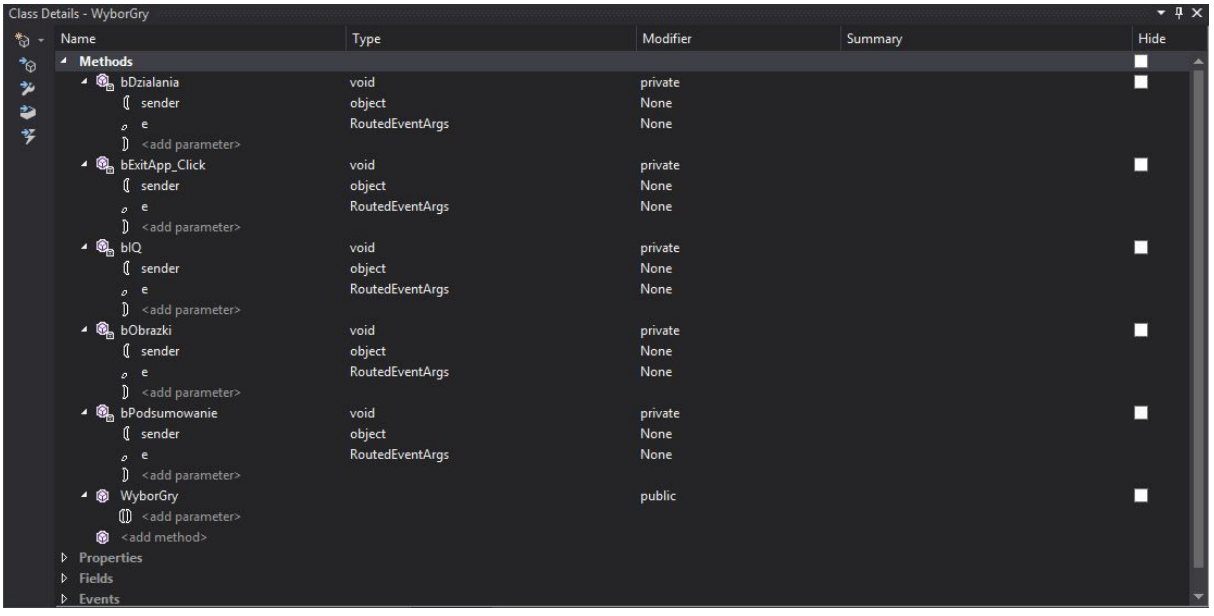
MODEL DANYCH

(pola, metody, właściwości, zdarzenia)

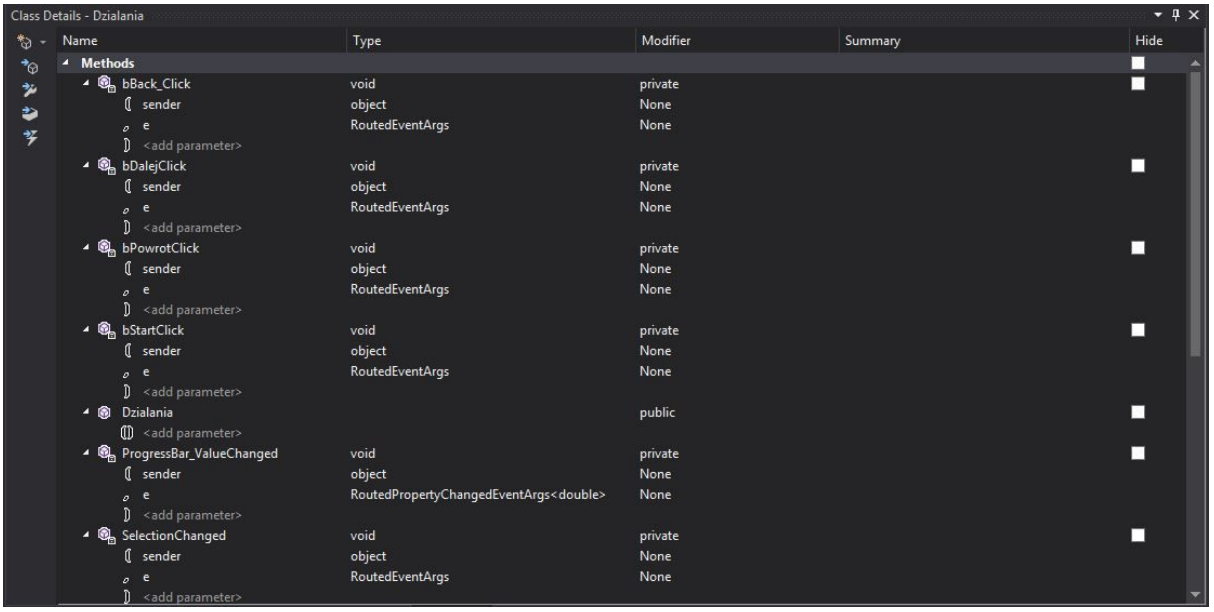
MainPage.xaml.cs

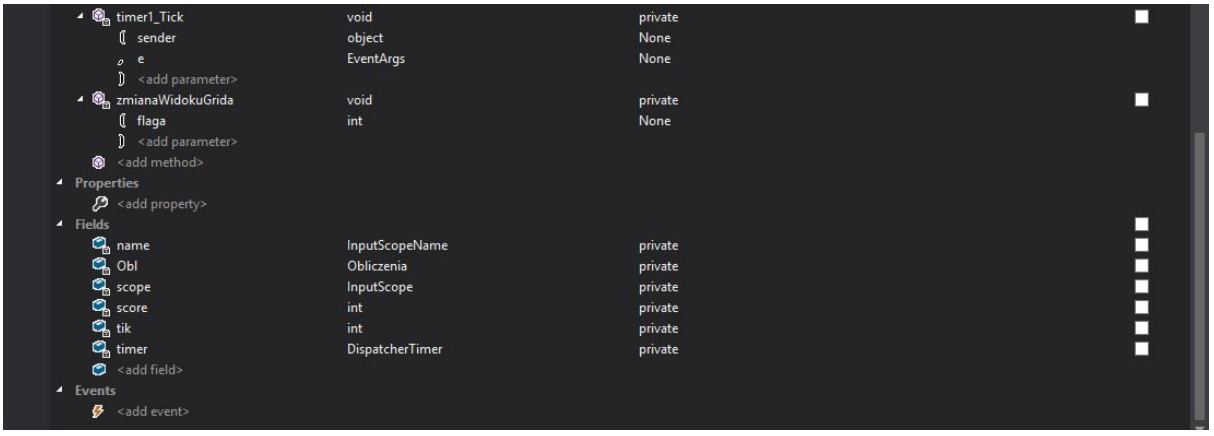


Wyborgry.xaml.cs

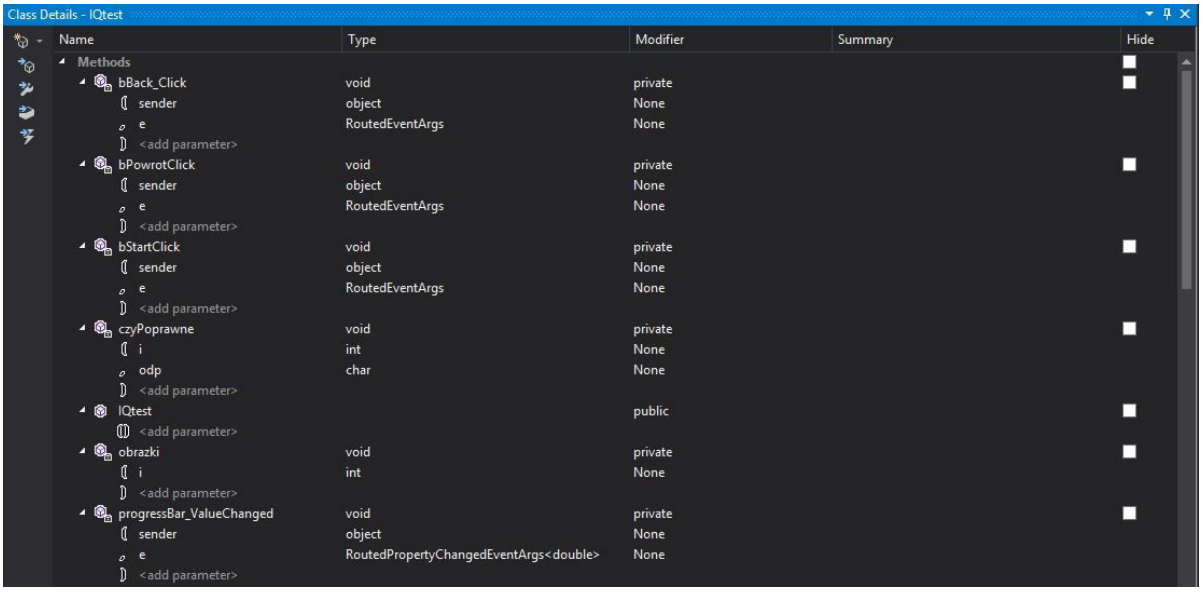


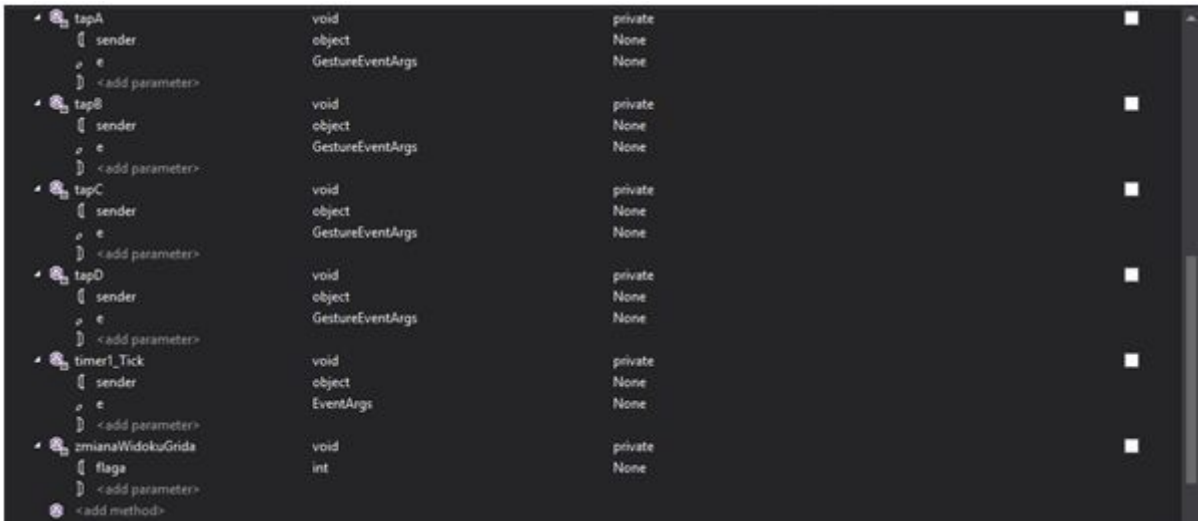
Dzialania.xaml.cs



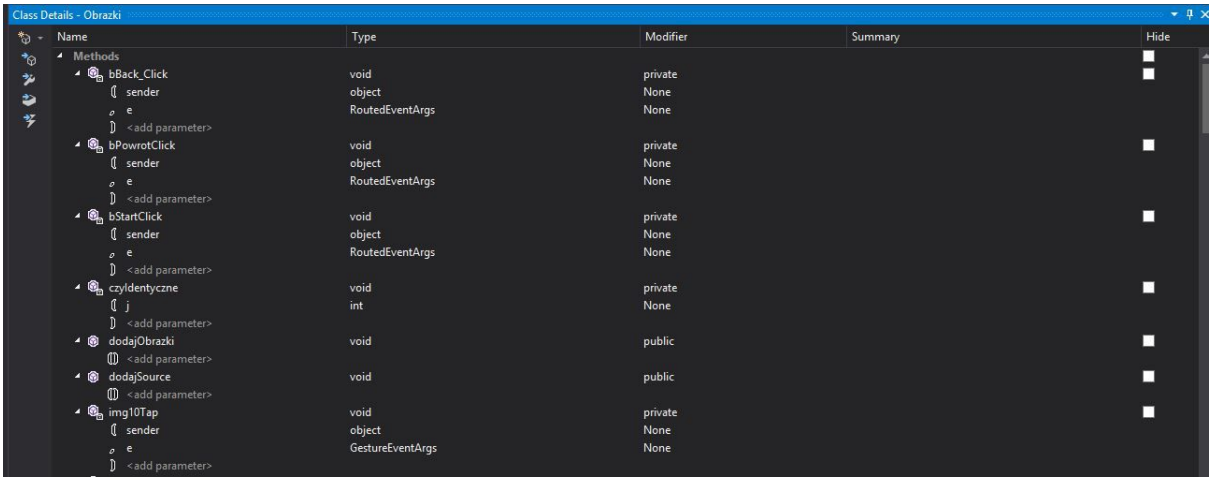


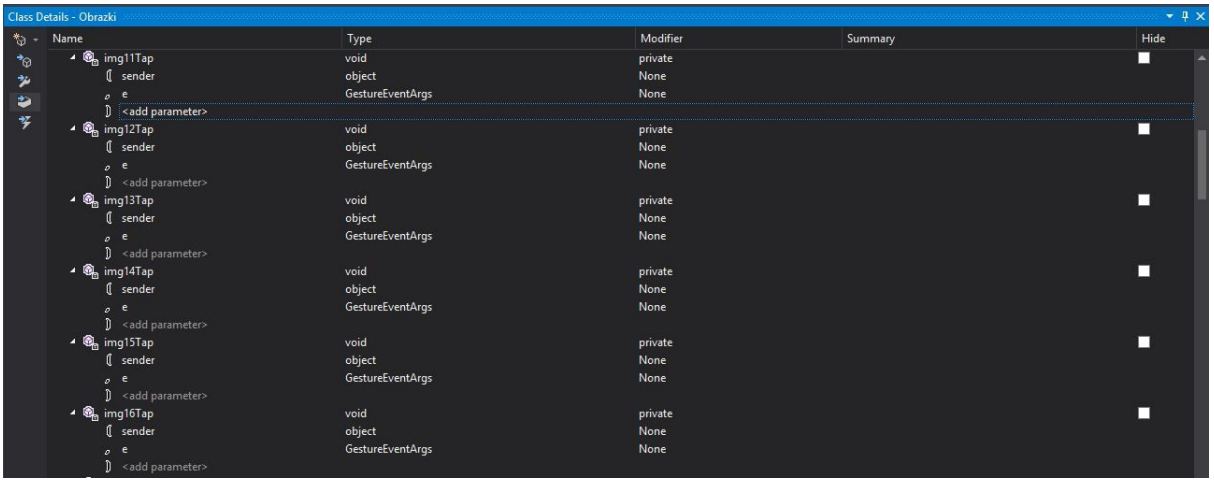
IQtest.xaml.cs

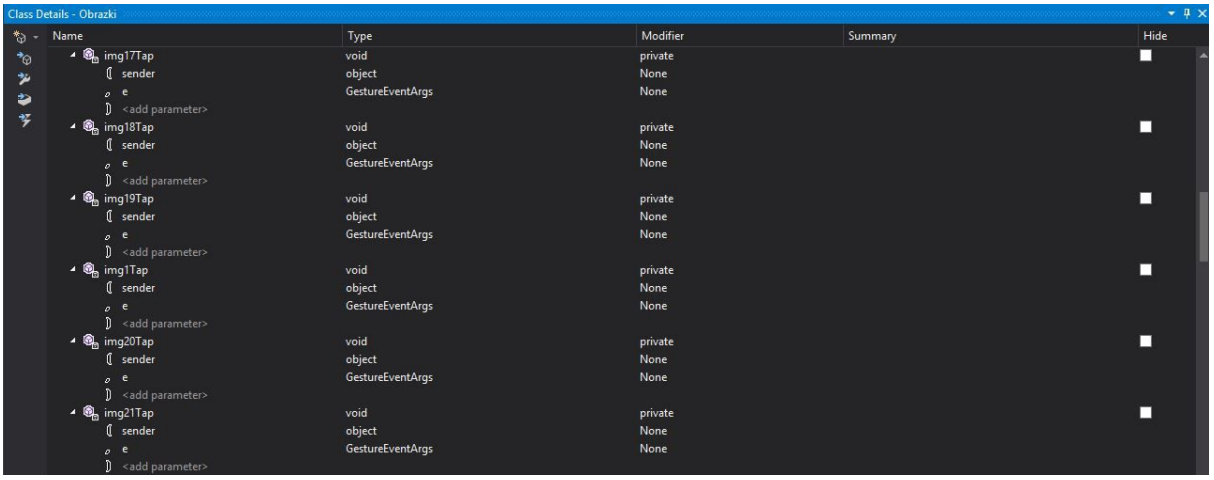


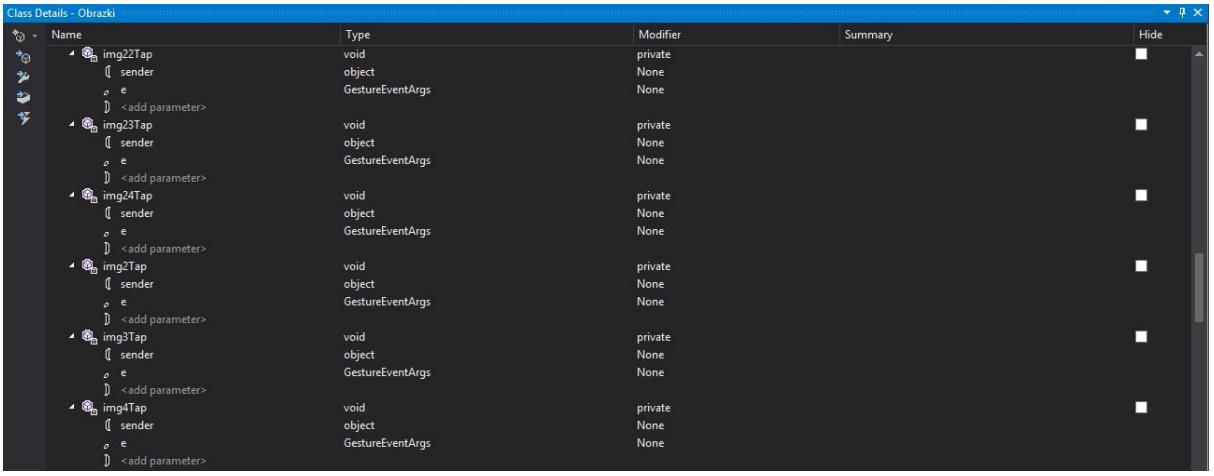


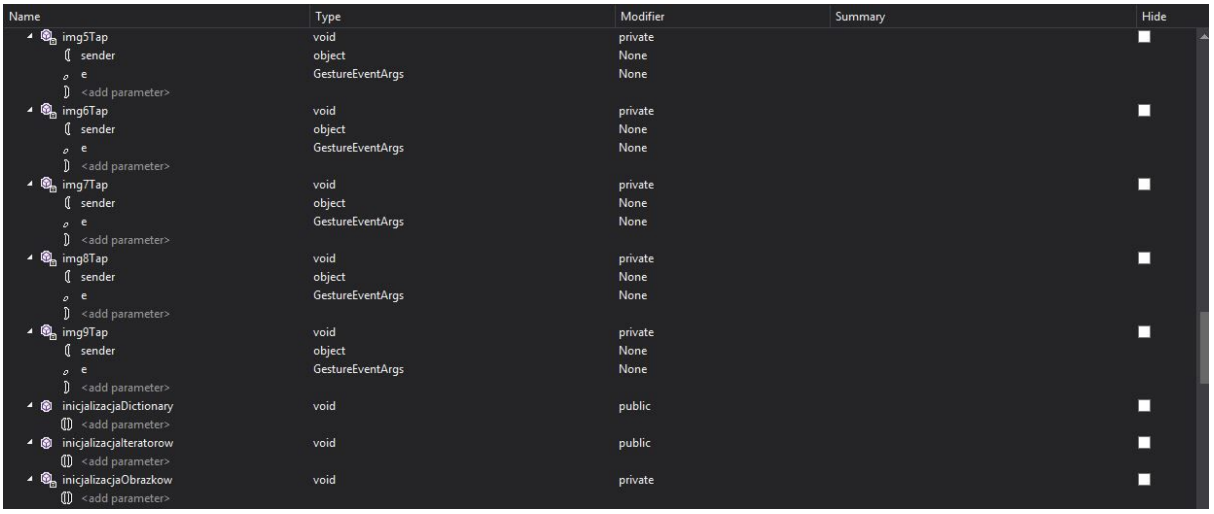
Obrazki.xaml.cs

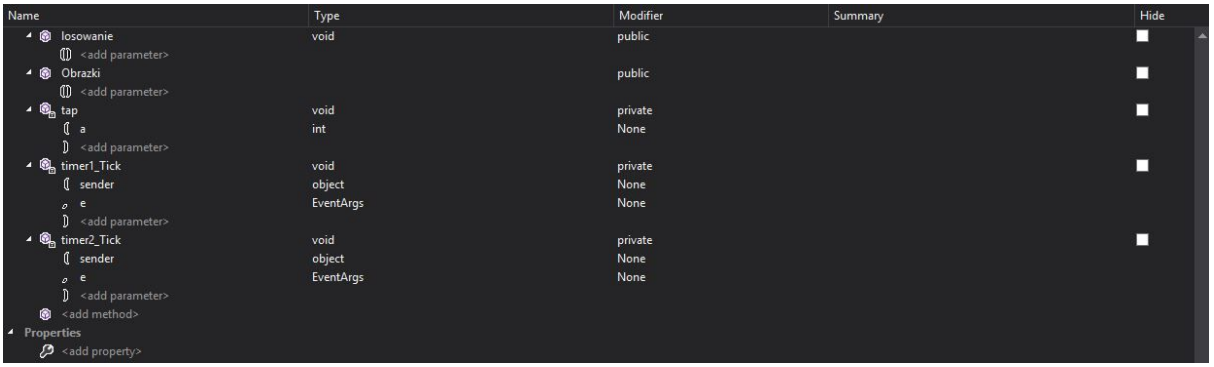






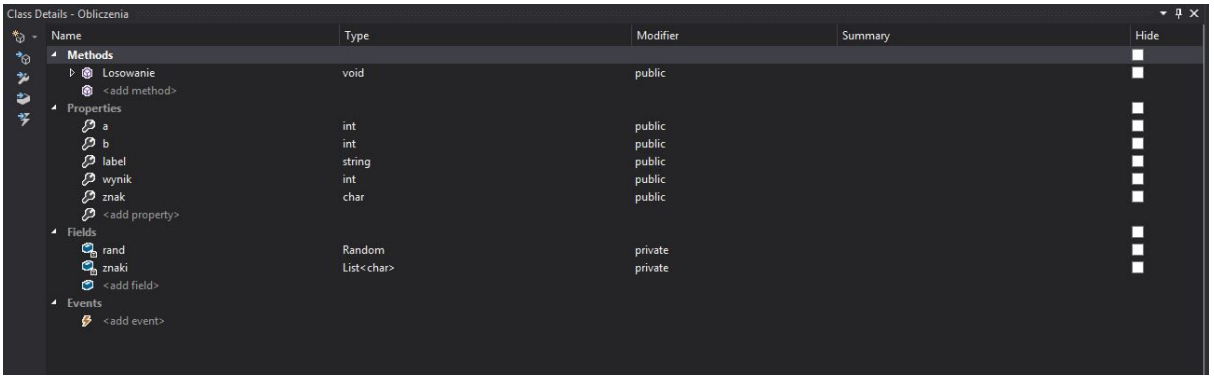

Class Details - Obrazki
Name
Type
Modifier
Summary
Hide
img22Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img23Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img24Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img2Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img3Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img4Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>


Name
Type
Modifier
Summary
Hide
img5Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img6Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img7Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img8Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
img9Tap
void
private
sender
object
None
e
GestureEventArgs
None
<add parameter>
inicjalizacjaDictionary
void
public
<add parameter>
inicjalizacjalteratorow
void
public
<add parameter>
inicjalizacjaObrazkow
void
private
<add parameter>

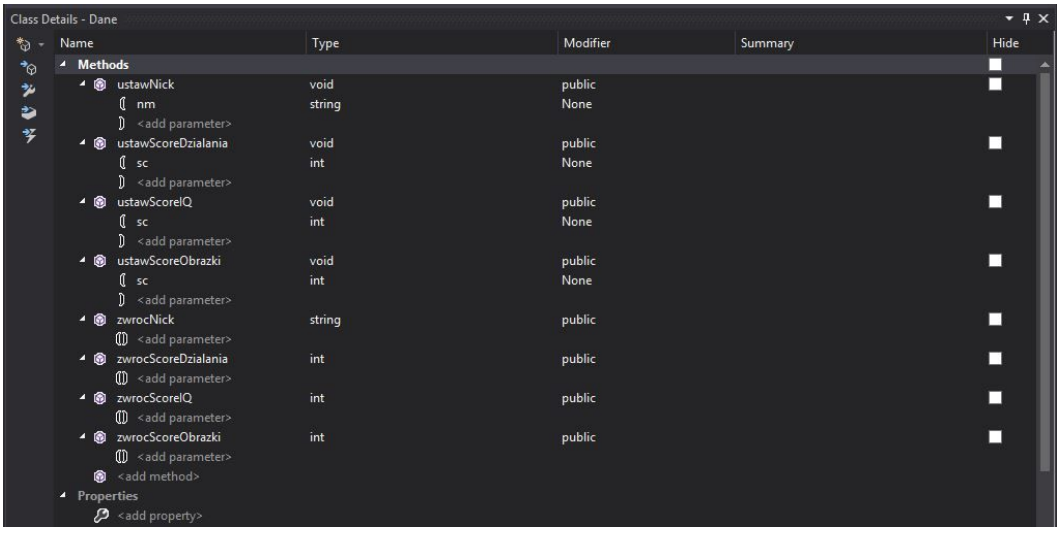

Name
Type
Modifier
Summary
Hide
losowanie
void
public
<add parameter>
Obrazki
public
<add parameter>
tap
void
private
a
int
None
<add parameter>
timer1_Tick
void
private
sender
object
None
e
EventArgs
None
<add parameter>
timer2_Tick
void
private
sender
object
None
e
EventArgs
None
<add parameter>
<add method>
Properties
<add property>

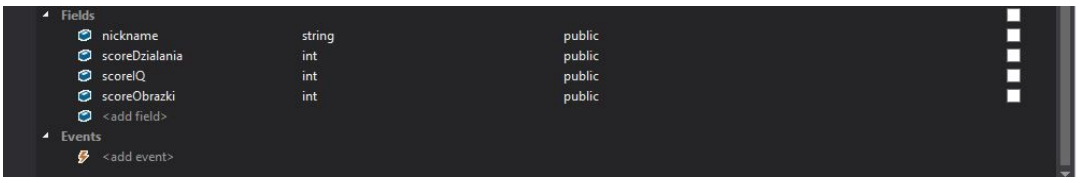
Podsumowanie.xaml.cs

Obliczenia.cs



Dane.cs





9. Opis klas i metod

MODEL DANYCH: TYPY I STRUKTURY DANYCH, MIEJSCA UŻYCIA

KLASA:

MainPage.xaml.cs -> główna klasa

ZMIENNE:

String nick - przechowuje nick użytkownika

KLASA:

Wyborgry.xaml.cs -> klasa w której użytkownik może wybrać konkurencję lub podsumowanie swoich osiągnięć w grze

ZMIENNE:

brak

KLASA:

Dzialanie.xaml.cs -> klasa odpowiadająca za rozwiązywanie działań na czas

ZMIENNE:

int tik - przechowywana jest pozostała wartość czasu który jest odmierzany

int score - przechowuje wynik gracza

string odpowiedz - przechowuje odpowiedz użytkownika

KLASA:

IQtest.xaml.cs ->klasa odpowiadająca za implementację testu IQ

ZMIENNE:

int score - przechowuje wynik gracza

int licznik - przechowuje wartość licznika, używanego w przełączaniu obrazków

int tik - przechowywana jest pozostała wartość czasu który jest odmierzany

STRUKTURA DANYCH:

char poprawna[] - przechowuje poprawne odpowiedzi

KLASA:

Obrazki.xaml.cs -> klasa odpowiadająca za zakrywanie i odkrywanie obrazków

ZMIENNE:

private static int licznik-pole służące do odkrycia maksymalnie dwóch obrazków

private static bool flaga-blokuje zwiększenie licznika bezpośrednio po sprawdzeniu czy obrazki są sobie równe

private static bool timerFlag-umożliwia wejście do intrukcji powodującej zakrycie tych samych obrazków

private static int przelacznik-po naciśnięciu odpowiedniego obrazka "czarny" obrazek zamieniany jest na obrazek "wylosowany"

private static int zakres- służy do zmniejszania zakresu losowania liczb

public static int score-przechowuje zdobyte punkty

private int tik2- służy do odmierzania czasu

private int tik1- czas odkrywania obrazków

STRUKTURY DANYCH:

public List<Image> obrazki-przechowuje obrazki

public List<BitmapImage> source- przechowywane są obiekty "uri" przechowujące z kolei bezpośrednią scieżkę do obrazka

public Dictionary<Image, BitmapImage> pictures-przechowuje obrazki

---

KLASA:

Podsumowanie.xaml.cs -> klasa przechowująca aktualne osiągnięcia użytkownika, z trzech etapów gier

ZMIENNE:

brak

KLASA:

Obliczenia.xaml.cs -> klasa która wykonuje obliczenia dla klasy działania

ZMIENNE:

int a - pierwsza liczba w równaniu

int b - druga liczba w równaniu

char znak - przechowuje znak równania

int wynik - przechowuje obliczony wynik równania

string label - przechowuje pełne równanie w formie stringa

int a1 - wylosowana pierwsza liczba od 1 do 100

int b1 - wylosowana druga liczba od 1 do 100

int idx1 - przechowuje wylosowany indeks znaku z listy znaków

STRUKTURY DANYCH:

List<char> - przechowuje listę znaków

KLASA:

Dane.xaml.cs -> klasa przechowująca dane o osiągnięciach użytkownika

ZMIENNE:

string nickname - przechowuje nick użytkownika

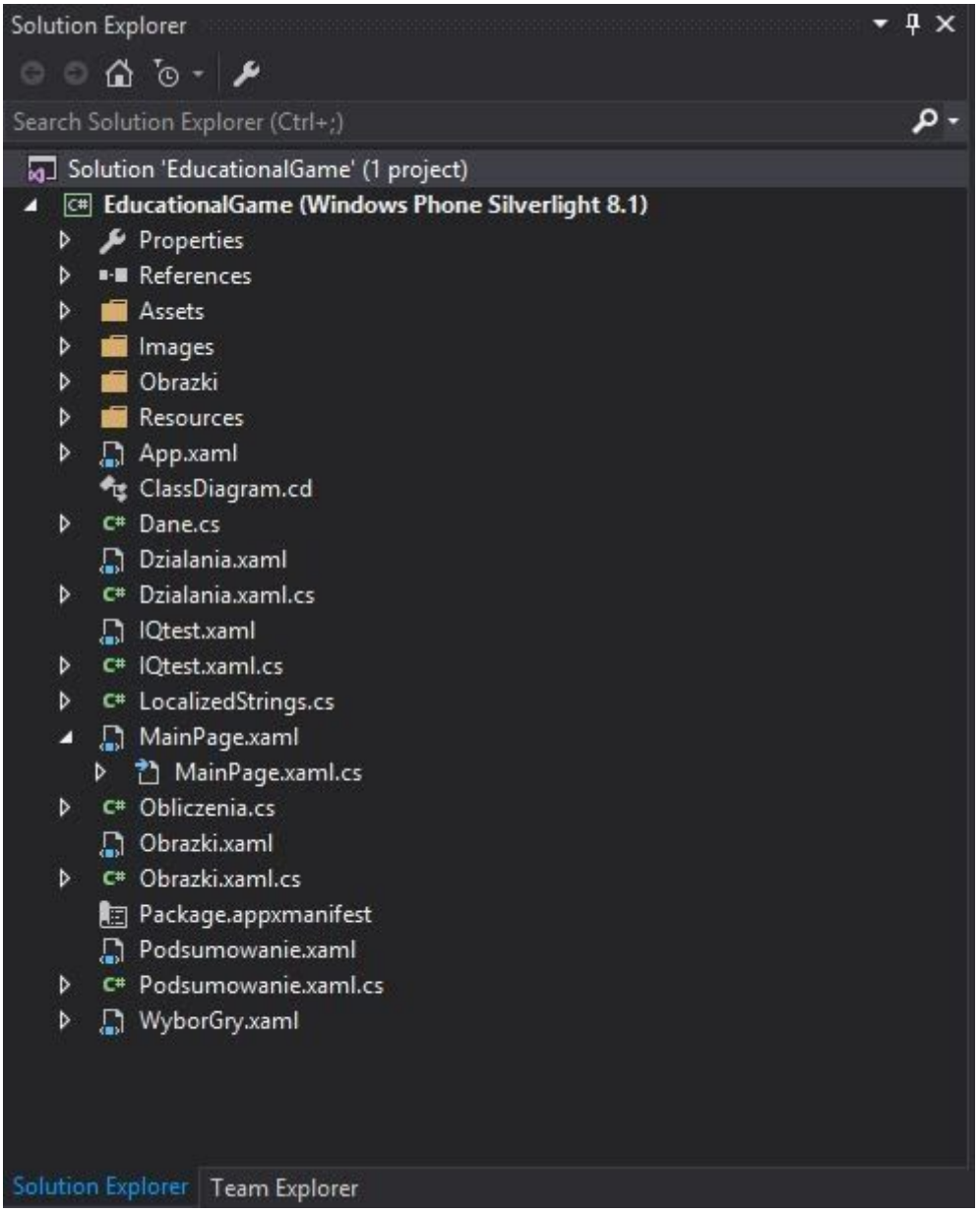
int scoreIQ - przechowuje wynik z gry IQTest

int scoreDziałania - przechowuje działania

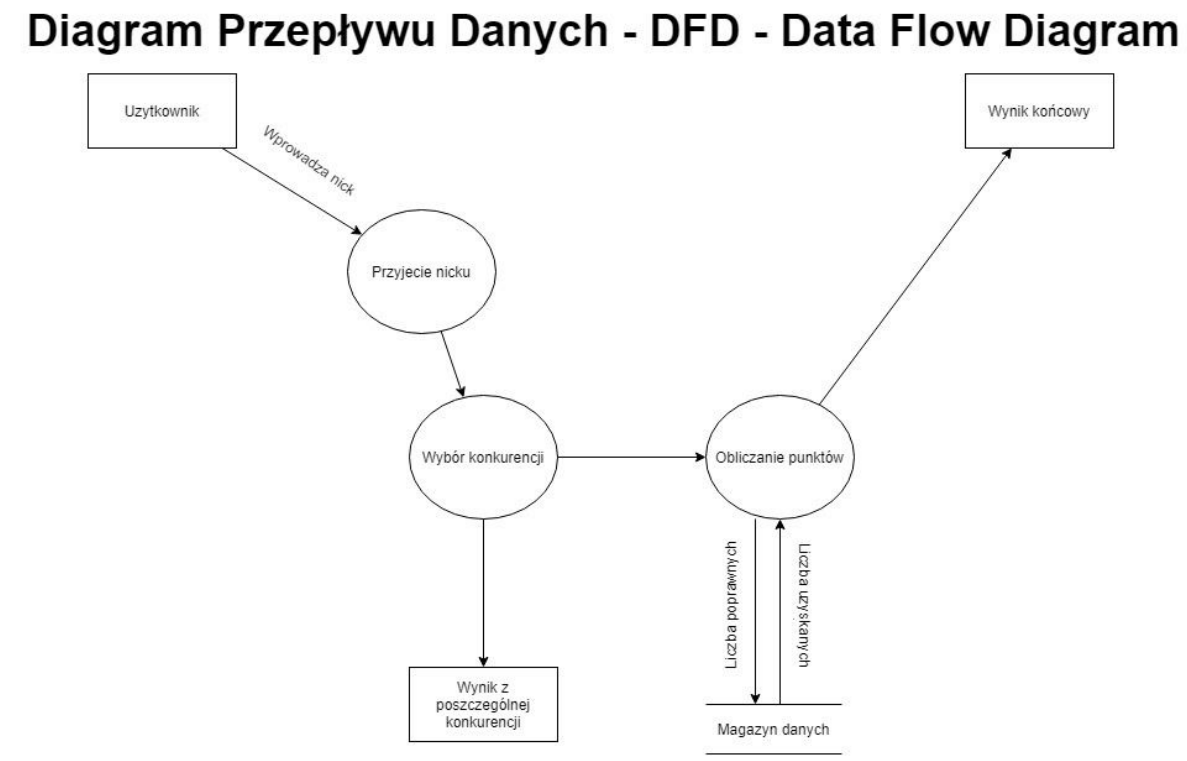
int scoreObrazki - przechowuje wyniki

KLASA	OPIS
MainPage.xaml.cs	Główna klasa aplikacji
Wyborgry.xaml.cs	Klasa, w której użytkownik może wybrać konkurencję lub obejrzeć rezultaty swoich osiągnięć w grze
Dzialania.xaml.cs	Klasa odpowiadająca za rozwiązywanie działań na czas
IQtest.xaml.cs	Klasa odpowiadająca za implementację testu IQ
Obrazki.xaml.cs	Klasa odpowiadająca za zakrywanie i odkrywanie obrazków klubów piłkarskich
Podsumowanie.xaml.cs	Klasa przechowująca aktualne osiągnięcia użytkownika, z trzech etapów gry
Obliczenia.cs	Klasa, która wykonuje obliczenia dla klasy działania
Dane.cs	Klasa przechowująca dane o osiągnięciach użytkownika

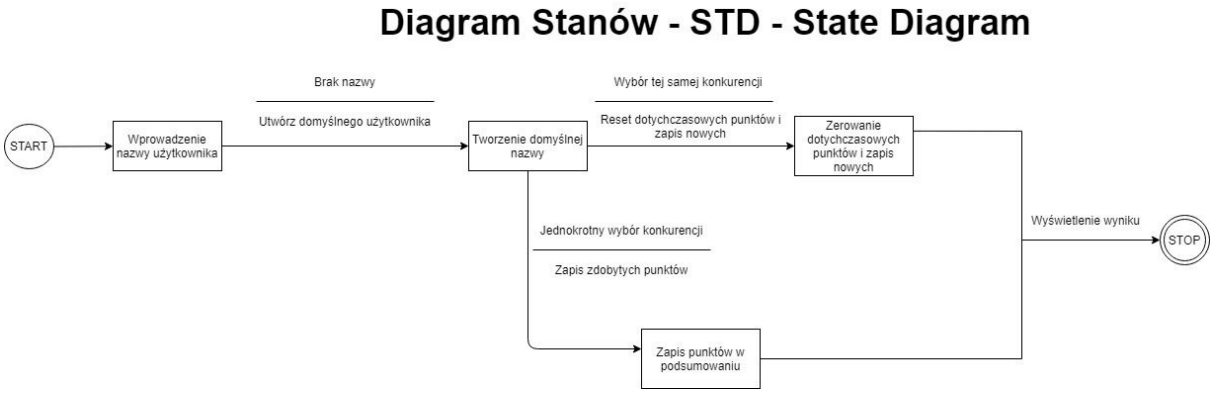
10. Drzewo katalogowo - plikowe projektu



11. Diagram przepływu danych - DFD - Data Flow Diagram



12. Diagram stanów - STD - State Diagram



13. Tutorial (wymagania oraz sposób obsługi wraz z zrzutami ekranu)

WYMAGANIA

- włączona funkcja systemu windows hyper-v
- microsoft visual studio community 2015
- windows phone 8.1 SDK

SPOSÓB OBSŁUGI

Celem aplikacji jest zweryfikowanie poziomu inteligencji gracza. Aplikacja nosi tytuł "EducationalGame". Umożliwia rozgrywkę w trzech następujących kategoriach: "Equations On Time - podstawy równań matematycznych", "Memory Game - prosta gra w zapamiętywanie par" oraz "IQ Test - popularny test inteligencji". W pierwszej konkurencji gracz sprawdzi swoje umiejętności szybkiego rozwiązywania równań matematycznych. W kolejnej kategorii użytkownik będzie musiał jak najszybciej odszukać wszystkie pary obrazków klubów piłkarskich z pośród dwunastu wszystkich par. W ostatniej dyscyplinie gracz musi wykazać się spostrzegawczością i umiejętnością logicznego myślenia, bowiem czeka na niego klasyczny test IQ. W celu utrudnienia rozgrywki gracz będzie zmagał się z największym rywalem, jakim jest nieubłagalnie upływający czas. Za każdą poprawną odpowiedź gracz nagradzany jest punktami. Po zakończeniu każdej z wyżej wymienionych kategorii, a także po zakończeniu całej rozgrywki, gracz otrzymuje możliwość podglądu dotychczas uzyskanych punktów, klikając w przycisk "Results".

ZRZUTY EKRANU


65 % 1 = ?
Next ->
<- Back
The End
Next ->
Discover All
Start ->
<- Back


2:26
MEMORY GAME
<- Back
2:27
MEMORY GAME
The End
Your Score: 3528
Next ->
2:28
IQ TEST
Prove Yourself
Click start button when you will be ready
Start ->
<- Back


2:28
IQ TEST
?
A
B
C
D
<- Back
2:29
IQ TEST
The End
Your Score: 57792
Next ->
2:30
RESULTS
Your nickname: default
Equations On Time
Score: 164280
Memory Game
Score: 3528
IQ Test
Score: 57792
Overall: 225600
<- Back