POLITECHNIKA
LUBELSKA
WYDZIAŁ MATEMATYKI
I INFORMATYKI TECHNICZNEJ

Kierunek: IAD

Dokumentacja do gry „Świat zwierząt”

Wykładowca:
dr. Paweł Wlaź

Autorzy:
Diana Kalyniak
Oleh Zemlianyi

Lublin

Spis treści

1 Cele projektu

Celem naszego projektu jest stworzenie gry w środowisku CodeBlocks z wykorzystaniem biblioteki WxWidgets. Postanowiliśmy stworzyć ciekawą grę związaną ze zwierzętami. Naszym celem jest nabycie umiejętności w obszarze korzystania z biblioteki WxWidgets do skutecznego tworzenia interfejsu użytkownika. Chcemy rozwijać umiejętności i wiedzę w zakresie języka programowania C++. Chcemy także doskonalić umiejętność pracy w zespole.

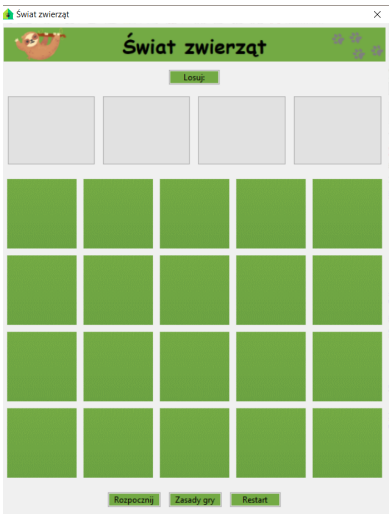
2 Opis gry

Stworzyliśmy grę **„Świat zwierząt”**, której zasadą jest odnalezienie wyglądu zwierzęcia po jego nazwie. Gra rozwija zrozumienie zwierząt, a ciekawy wygląd przyciąga uwagę użytkowników. Cały interfejs wykonany jest w jednym kolorze – zielonym (rys. 1). Zaraz za nazwą gry znajduje się przycisk **Losuj** (rys. 2) zmieniający obrazek z wyglądem zwierząt, ma to na celu umożliwienie zmiany obrazka jeśli użytkownik nie może go od razu znaleźć. Na dole są 3 przyciski: **Rozpocznij, Zasady gry, Restart** (rys. 3). W środku po naciśnięciu przycisku Rozpocznij zostaną narysowane 4 nazwy zwierząt i 20 obrazków z ich wyglądem(rys. 4).

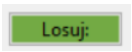
3 Zasady gry

Zasady gry są bardzo proste i są wyświetlane po naciśnięciu przycisku **Zasady gry**(rys. 5). Po kliknięciu przycisku **Rozpocznij** gra się rozpoczyna i musisz znaleźć dopasowanie między nazwami i wyglądem zwierząt. Musisz kliknąć tylko obrazy z wyglądem(rys. 8). Jeśli dwukrotnie klikniesz niewłaściwy obrazek, wyświetli się komunikat i gra się rozpocznie od nowa(rys. 6). Jeśli wybrano 4 prawidłowe obrazki, gra się zakończy i wyświetli się wiadomość powitalna(rys. 7) .

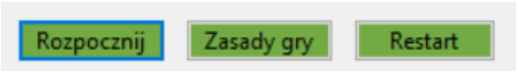
4 Wygląd gry



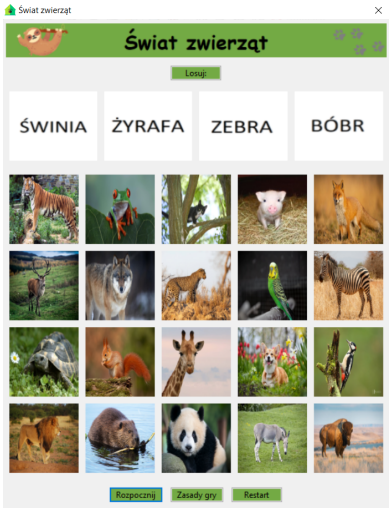
Rysunek 1: Wygląd gry po uruchomieniu



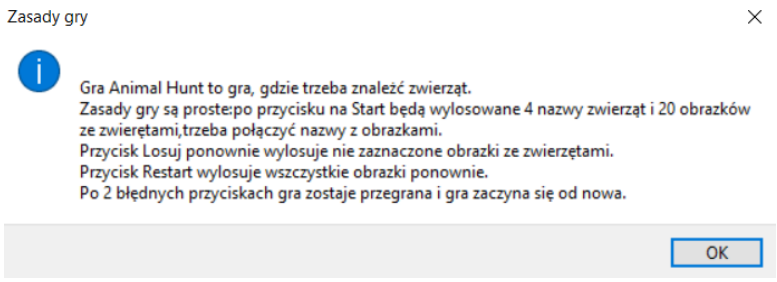
Rysunek 2: Losuj



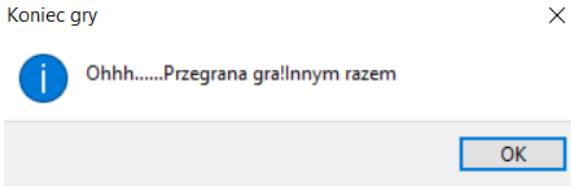
Rysunek 3: Przyciski na dole



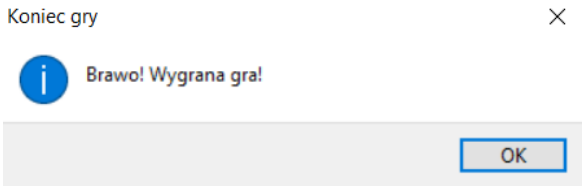
Rysunek 4: Wygląd gry po przycisku Rozpocznij



Rysunek 5: Zasady gry



Rysunek 6: Komunikat o przegranej grze



Rysunek 7: Komunikat o wygranej grze

Rysunek 8: Dobra i zła odpowiedź

5 Kod gry

Mamy 12 fragmentów kodu z wyjaśnieniem. Pierwszy punkt jest plik.h, następne 4 są napisane w głównym konstruktorze.

1. Deklaracja tablicy wskaźników do 4 obiektów typu wxBitmapButton, która będzie przechowywać obrazki nazw zwierząt. Deklaracja funkcji ResizeImageToButton, która będzie odpowiedzialna za dostosowanie rozmiaru obrazka do rozmiaru przycisku. Deklaracja tablicy wskaźników do 20 obiektów typu wxBitmapButton, która przechowuje obrazki z wyglądem zwierząt. Deklaracje zmiennych przechowujących ścieżkę do folderu z obrazkami, rozszerzenie plików obrazków oraz wektor nazw zwierząt. Podobne deklaracje do poprzednich, ale dotyczące innego zestawu obrazków (20 obrazków do wybierania). Deklaracja przycisku, który służy do wyświetlenia informacji o grze. Deklaracje dwóch obiektów typu wxBitmap przechowujących obrazy „OK" i „Nie OK". Deklaracje wektorów przechowujących nazwy wylosowanych zwierząt oraz zbiór liczb całkowitych służący do przechowywania indeksów „zakrytych" przycisków. Deklaracje dwóch zmiennych przechowujących liczbę błędów i liczbę poprawnych odpowiedzi w grze. Deklaracja zmiennej logicznej (bool), która informuje, czy gra została rozpoczęta.

```
    wxBitmapButton* animalButtons[4];

    void ResizeImageToButton(wxImage& image, wxBitmapButton* button);

    wxBitmapButton* pola[20];

     wxString imageDirectory;
     wxString imageExtension;
     vector<wxString> animalNames;

    wxString imageDirectory1;
    wxString imageExtension1;
    vector<wxString> animalNames1;

    wxButton* aboutButton;

    wxBitmap obrazOk;
    wxBitmap obrazNieOk;

    vector<wxString> Wylosowane4;
    vector<wxString> Wylosowane20;
    set<int> zakryte;

    int liczbaBledow;
    int liczbaOk;

    bool gameStarted;
```

2. Pętla for służy do tworzenia tablicy przycisków (wxBitmapButton). Przyciski są tworzone na podstawie obrazu z pliku „kwadrat.png" o rozmiarze 100x100 pikseli. Przypisuje pierwszy element tablicy pola do BitmapButton7. Poniższe linie kodu tworzą tablicę animalButtons, która zawiera wskaźniki do przycisków o określonych identyfikatorach

```

  for(int i=1; i<20; i++)
  {
    pola[i] = new wxBitmapButton(this, wxNewId(), wxBitmap(wxImage(_T("kwadrat.png"))),
        wxDefaultPosition, wxSize(100,100), wxBU_AUTODRAW, wxDefaultValidator,
        _T("ID_BITMAPBUTTON7"));
    FlexGridSizer2->Add(pola[i], 1, wxALL|wxALIGN_CENTER_HORIZONTAL|wxALIGN_CENTER_VERTICAL,
        5);
    Connect(pola[i]->GetId(),wxEVT_COMMAND_BUTTON_CLICKED,
    (wxObjectEventFunction)&Animal_game_classworkDialog::
    OnBitmapButton7Click);
  }
  pola[0] = BitmapButton7;



  animalButtons[0] = dynamic_cast<wxBitmapButton*>(FindWindowById(ID_BITMAPBUTTON3));
```

```
15  animalButtons[1] = dynamic_cast<wxBitmapButton*>(FindWindowById(ID_BITMAPBUTTON4));
16  animalButtons[2] = dynamic_cast<wxBitmapButton*>(FindWindowById(ID_BITMAPBUTTON5));
17  animalButtons[3] = dynamic_cast<wxBitmapButton*>(FindWindowById(ID_BITMAPBUTTON6));
18  animalButtons[4] = dynamic_cast<wxBitmapButton*>(FindWindowById(ID_BITMAPBUTTON7));
```

3. Obrazki, które wskazują, że wybrane zdęcie jest poprawne lub nie.

```
1  obrazOk = wxBitmap(wxImage("ok.png"));
2  obrazNieOk = wxBitmap(wxImage("nie_ok.png"));
3  if (!obrazOk.IsOk() || !obrazNieOk.IsOk()) {
4    wxMessageBox("Nie mozna zaladowac obrazkow: ok.png lub nie_ok.png", "Blad", wxOK |
         wxICON_ERROR, this);
5  }
```

4. Wczytywanie wszystkich obrazków z folderów zdj i zdj2.

```
1  imageDirectory = "zdj/";
2  imageExtension = ".png";
3
4  animalNames = {"bobr", "dzieciol", "jelen", "kot", "krol",
5    "kurczak", "lampart", "lew", "lis", "niedzwiedz",
6    "panda", "papuga", "pies", "slon", "swinia",
7    "tygrys", "zaba", "zebra", "zolw", "zyrafa"};
8
9  imageDirectory1 = "zdj2/";
10 imageExtension1 = ".png";
11 animalNames1 = {"bobr", "dzieciol", "jelen", "kot", "krol",
12   "kurczak", "lampart", "lew", "krowa", "lis", "niedzwiedz",
13   "panda", "papuga", "pies", "slon", "swinia",
14   "tygrys", "zaba", "zebra", "zolw", "zyrafa", "wilk","osiol","wiewiorka","bizon"};
```

5. Stworzenie zmiennych, jakie odpowiadają za liczenie poprawnych i nie poprawnych odpowiedzi. Ustawienie zmiennej kontrolującej, czy naciśnięto przycisk rozpocznij na false na początku.

```
1    liczbaBledow = 0;
2    liczbaOk = 0;
3    gameStarted=false;
```

6. Dopasowanie rozmiarów obrazków.

```
1    void Animal_game_classworkDialog::ResizeImageToButton(wxImage& image, wxBitmapButton*
         button)
2    {
3      wxSize buttonSize = button->GetSize();
4      image.Rescale(buttonSize.GetWidth(), buttonSize.GetHeight(), wxIMAGE_QUALITY_HIGH);
5    }
```

7. Używanie generatora liczb ranlux48.

```
1    ranlux48 gen(time(0));
```

8. Funkcja OnButton1Click jest obsługą zdarzenia kliknięcia przycisku „Rozpocznij”. Sprawdza, czy gra już została rozpoczęta. Jeśli tak, to funkcja nie wykonuje dalszych operacji i kończy się. Potem przetasowuje wektory animalNames1 i animalNames przy użyciu generatora liczb losowych gen. Czyści wektory Wylosowane4 i Wylosowane20, które przechowują nazwy wylosowanych zwierząt. Uzycie pierwszych czterech indeksów po przetasowaniu jako wybrane nazwy zwierząt. Uzycie pierwszych 20 indeksów po przetasowaniu jako wybrane obrazy zwierząt. Resetuje liczniki błędów i poprawnych odpowiedzi.

```
1   void Animal_game_classworkDialog::OnButton1Click(wxCommandEvent& event)
2   {
3     if (gameStarted) {
4       return;
5     }
6     gameStarted = true;
7
8     shuffle(animalNames1.begin(), animalNames1.end(), gen);
9     shuffle(animalNames.begin(), animalNames.end(), gen);
10
11    Wylosowane4.clear();
12    Wylosowane20.clear();
13
14    for (int i = 0; i < 4; ++i) {
15      wxString imagePath = imageDirectory + animalNames[i] + imageExtension;
16      wxImage image;
17
18      if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
19        //Dopasowanie obrazkow(rozmiary)
20        ResizeImageToButton(image, animalButtons[i]);
21        wxBitmap bitmap(image);
22        animalButtons[i]->SetBitmap(bitmap);
23        animalButtons[i]->Show();
24
25        // Dodanie do wektora Wylosowane4 (tylko nazwa)
26        Wylosowane4.push_back(animalNames[i]);
27      } else {
28        wxMessageBox("Nie mozna zaladowac obrazka: " + imagePath, "Blad", wxOK |
              wxICON_ERROR, this);
29      }
30    }
31
32    Layout();
33
34    for (int i = 0; i < 20; ++i) {
35      wxString imagePath = imageDirectory1 + animalNames1[i] + imageExtension1;
36      wxImage image;
37      if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
38        //Dopasowanie obrazkow(rozmiary)
39        ResizeImageToButton(image, pola[i]);
40        wxBitmap bitmap(image);
41        pola[i]->SetBitmap(bitmap);
42        pola[i]->Show();
43        Wylosowane20.push_back(animalNames1[i]);
44      } else {
45        wxMessageBox("Nie mozna zaladowac obrazka: " + imagePath, "Blad", wxOK |
              wxICON_ERROR, this);
46      }
47    }
48    liczbaBledow = 0;
49    liczbaOk = 0;
50    Layout();
51    Refresh();
52    }
```

9. Przycisk LOSUJ, ładowanie i przypisanie obrazków do przycisków z folderu zdj2. Przycisk jest nieaktywny do momentu naciśnięcia przycisku Rozpocznij.

```
void Animal_game_classworkDialog::OnButton3Click(wxCommandEvent& event)
{
  if (!gameStarted) {
    return;
  }
  Wylosowane20.clear();

  shuffle(animalNames1.begin(), animalNames1.end(), gen);

  for (int i = 0; i < 20; ++i) {
    if (zakryte.find(pola[i]->GetId()) != zakryte.end())
    continue;
    wxString imagePath = imageDirectory1 + animalNames1[i] + imageExtension1;
    wxImage image;
    if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
      ResizeImageToButton(image, pola[i]);
      wxBitmap bitmap(image);
      pola[i]->SetBitmap(bitmap);
      pola[i]->Show();
      Wylosowane20.push_back(animalNames1[i]);
    }
  }
  Layout();
}
```

10. Przycisk Zasady gry, jaki wyświetla zasady gry.

```
//About
void Animal_game_classworkDialog::OnButton2Click(wxCommandEvent& event)
{
  wxString aboutMessage = wxT("\n Gra Animal Hunt to gra, gdzie trzeba znalezc zwierzat.
      \n Zasady gry sa proste:po przycisku na Start beda wylosowane 4 nazwy zwierzat i 20
      obrazkow \n ze zwieretami,trzeba polaczyc nazwy z obrazkami. \n Przycisk Losuj
      ponownie wylosuje nie zaznaczone obrazki ze zwierzetami. \n Przycisk Restart
      wylosuje wszczystkie obrazki ponownie. \n Po 2 blednych przyciskach gra zostaje
      przegrana i gra zaczyna sie od nowa.");
  wxMessageBox(aboutMessage, wxT("Zasady gry"), wxOK | wxICON_INFORMATION, this);
}
```

11. Przycisk RESTART. Przycisk jest nieaktywny do momentu naciśnięcia przycisku Rozpocznij. Ladowanie i przypisanie obrazkow do przyciskow z folderu zdj2 i zgj. Kod jest podobny do tego w przycisku Rozpocznij, ale przycisk Restart losuje obrazki wiele razy, a Rozpocznij tylko raz na początku gry.

```
void Animal_game_classworkDialog::OnButton4Click(wxCommandEvent& event)
{
  if (!gameStarted) {
    return;
  }
  Wylosowane20.clear();
  zakryte.clear();

  shuffle(animalNames1.begin(), animalNames1.end(), gen);

  for (int i = 0; i < 20; ++i) {
    wxString imagePath = imageDirectory1 + animalNames1[i] + imageExtension1;
    wxImage image;
    if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
      //Dopasowanie obrazkow(rozmiary)
      ResizeImageToButton(image, pola[i]);
      wxBitmap bitmap(image);
      pola[i]->SetBitmap(bitmap);
      pola[i]->Show();
      Wylosowane20.push_back(animalNames1[i]);
    } else {
      wxMessageBox("Nie mozna zaladowac obrazka: " + imagePath, "Blad", wxOK |
          wxICON_ERROR, this);
    }
  }
```

```
25      Layout();
26
27      wxString imageDirectory = "zdj/";
28      wxString imageExtension = ".png";
29
30      for (int i = 0; i < 4; ++i) {
31        wxString imagePath = imageDirectory + animalNames[i] + imageExtension;
32        wxImage image;
33        if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
34        //Dopasowanie obrazkow(rozmiary)
35          ResizeImageToButton(image, animalButtons[i]);
36          wxBitmap bitmap(image);
37          animalButtons[i]->SetBitmap(bitmap);
38          animalButtons[i]->Show();
39          Wylosowane4.push_back(animalNames[i]);
40        } else {
41        wxMessageBox("Nie mozna zaladowac obrazka: " + imagePath, "Blad", wxOK |
              wxICON_ERROR, this);
42        }
43      }
44
45      liczbaBledow = 0;
46      liczbaOk = 0;
47      Layout();
48      Refresh();
49    }
```

12. Sprawdzenie, czy klikalny obraz został poprawnie wybrany przez użytkownika. Pobieramy ID kliknięgo przycisku. Potem sprawdźenie, czy zwierzę z jednego wektoru są takie same jak i w drugim wektorze. Jeżeli tak to zmiana obrazka na „ok.png" i zmiana zmiennej, który liczy prawidłowe odpowiedzi, w przeciwnym razie zmiana obrazu na „nie-ok.png" i zmiana zmiennej zliczającej fałszywe kliknięcia. Ładowanie i przypisanie obrazków do przycisków z folderu zdj2 i dopasowanie rozmiarów obrazkow. Użycie pierwszych 4 indeksów po przetasowaniu jako wybrane obrazy i dopasowanie rozmiarów obrazkow. Potem użycie pierwszych 20 indeksów po przetasowaniu jako wybrane obrazy wyglądu zwierząt. Jeżeli 2 obrazki są „nie-ok", wyświetlamy komunikat o przegranej gre. Jeżeli 4 obrazki są „ok", wyświetlamy komunikat o wygranej gre.

```
1     void Animal_game_classworkDialog::OnBitmapButton7Click(wxCommandEvent& event)
2     {
3       wxBitmapButton* clickedButton = dynamic_cast<wxBitmapButton*>(event.GetEventObject());
4       if (!clickedButton) {
5         return;
6       }
7
8       if (!gameStarted) {
9         return;
10      }
11
12
13      int clickedButtonID = clickedButton->GetId();
14
15      if (zakryte.find(clickedButtonID) != zakryte.end())
16      return;
17
18      wxString clickedAnimalName;
19      for (int i = 0; i < 20; ++i) {
20        if (pola[i]->GetId() == clickedButtonID) {
21          clickedAnimalName = animalNames1[i];
22          break;
23        }
24      }
25
26
27      if (find(Wylosowane4.begin(), Wylosowane4.end(), clickedAnimalName) !=
            Wylosowane4.end()) {
28        // Zmiana obrazka na "ok.png"
29        clickedButton->SetBitmap(obrazOk);
30        zakryte.insert(clickedButtonID);
31        liczbaOk++;
32      } else {
```

```
33          // Zmiana obrazka na "nie_ok.png"
34          clickedButton->SetBitmap(obrazNieOk);
35          zakryte.insert(clickedButtonID);
36          liczbaBledow++;
37
38          if (liczbaBledow >= 2) {
39            wxMessageBox("Ohhh.....Przegrana gra!Innym razem", "Koniec gry", wxOK |
                  wxICON_INFORMATION, this);
40            Wylosowane20.clear();
41            zakryte.clear();
42            shuffle(animalNames1.begin(), animalNames1.end(), gen);
43
44
45            for (int i = 0; i < 20; ++i) {
46              wxString imagePath = imageDirectory1 + animalNames1[i] + imageExtension1;
47              wxImage image;
48              if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
49                ResizeImageToButton(image, pola[i]);
50                wxBitmap bitmap(image);
51                pola[i]->SetBitmap(bitmap);
52                pola[i]->Show();
53                Wylosowane20.push_back(animalNames1[i]);
54              } else {
55                wxMessageBox("Nie ma  obrazka: " + imagePath, "Blad", wxOK | wxICON_ERROR,
                      this);
56              }
57            }
58            Layout();
59
60
61            wxString imageDirectory = "zdj/";
62            wxString imageExtension = ".png";
63
64            Wylosowane4.clear();
65            shuffle(animalNames.begin(), animalNames.end(), gen);
66
67            for (int i = 0; i < 4; ++i) {
68              wxString imagePath = imageDirectory + animalNames[i] + imageExtension;
69              wxImage image;
70              if (image.LoadFile(imagePath, wxBITMAP_TYPE_PNG)) {
71                ResizeImageToButton(image, animalButtons[i]);
72                wxBitmap bitmap(image);
73                animalButtons[i]->SetBitmap(bitmap);
74                animalButtons[i]->Show();
75                Wylosowane4.push_back(animalNames[i]);
76              } else {
77                wxMessageBox("Nie mozna zaladowacc obrazka: " + imagePath, "Blad", wxOK |
                      wxICON_ERROR, this);
78              }
79            }
80            liczbaOk=0;
81            liczbaBledow=0;
82            Layout();
83
84            Refresh();
85          }
86
87        }
88
89
90        if (liczbaOk == 4) {
91          wxMessageBox("Brawo! Wygrana gra!", "Koniec gry", wxOK | wxICON_INFORMATION, this);
92          EndModal(wxID_CANCEL);  }
93        }
```

6 Podsumowanie

- Celem projektu było stworzenie gry edukacyjnej w języku C++ przy użyciu biblioteki WxWidgets. Głównymi założeniami były rozwijanie umiejętności w obszarze interfejsu użytkownika oraz doskonalenie umiejętności programowania i pracy zespołowej.
- Gra „Świat zwierząt" umożliwia graczom odnajdywanie zwierząt na podstawie ich nazw. Interfejs gry utrzymany jest w jednolitym kolorze, a losowe zmiany obrazków dodają elementu losowości i ciekawości.
- Proste zasady gry polegają na dopasowywaniu nazw zwierząt do ich obrazków. Błędne wybory powodują informację o błędzie, a poprawne odpowiedzi kończą grę sukcesem.
- Starannie zaprojektowany interfejs gry zapewnia przejrzystość i czytelność. Elementy graficzne są estetyczne, co ułatwia korzystanie z aplikacji.
- Kod napisany zgodnie z dobrymi praktykami programowania w języku C++. Wykorzystanie biblioteki WxWidgets umożliwiło efektywne zarządzanie interfejsem użytkownika.

7 Podział pracy

Wymyślanie pomysłów na projekt, wybieranie zdjęć i wymyślanie nazwy gry: **Oleh Zemlianyi**.

Wygląd graficzny programu: **Diana Kalyniak**.

Cały kod gry i kontrola błędów kodu: **Diana Kalyniak 60% i Oleh Zemlianyi 40%**.

Dokumentacja: **Oleh Zemlianyi 50% i Diana Kalyniak 50%**.

Sprawdzanie i kontrola estetyki dokumentacji: **Oleh Zemlianyi**.