

Kombinatoryczna teoria liczb

Opis projektu

*Anna Zawadzka
Piotr Waszkiewicz
Przemysław Rząd*

4 listopada 2016

Cel projektu

Projekt ma na celu zaimplementowanie gry w **rozdzielanie Szemerediego**.

Gra odbywa się w turach. Gracz mierzy się z komputerem, dla którego zaimplementowana zostanie strategia gry.

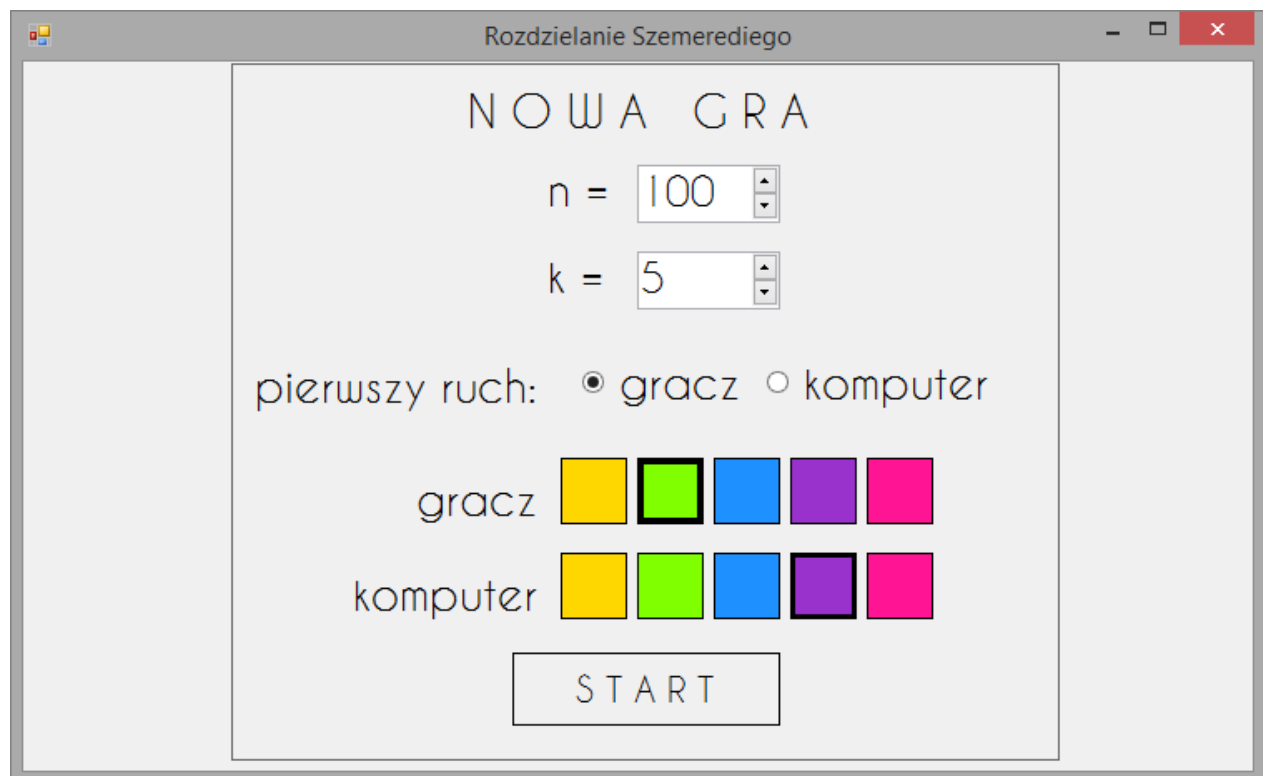
Dla wybranych wartości liczb **n** i **k**, runda polega na:

- wyborze dwóch (dotąd niewybranych) liczb ze zbioru $[n]$ przez gracza pierwszego
- Wyborze jednej ze wskazanych liczb przez gracza drugiego, która zostanie pokolorowana na jego kolor. Druga ze wskazanych liczb kolorowana jest na kolor gracza pierwszego

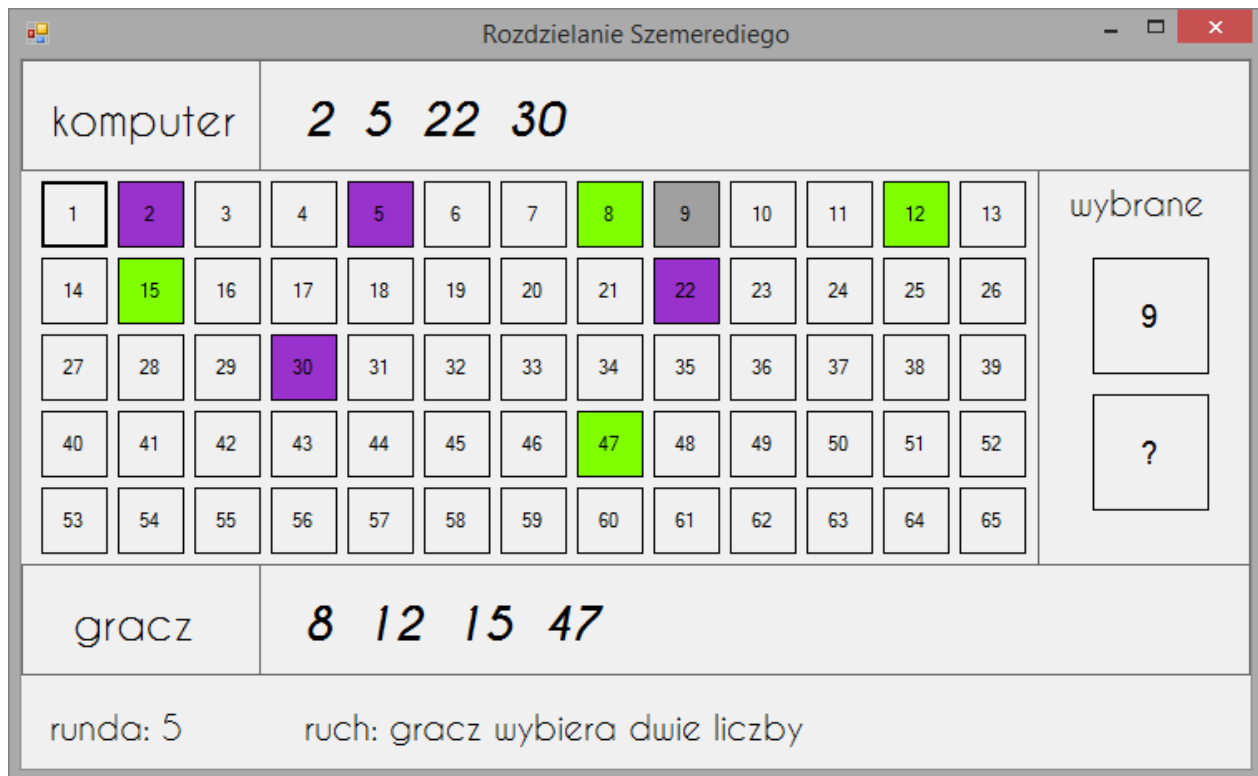
W rundach nieparzystych role się zamieniają. Wygrywa gracz, który pierwszy będzie miał **k**-elementowy ciąg arytmetyczny w swoim kolorze.

Graficzny interfejs użytkownika

Po uruchomieniu programu zostanie wyświetlone okno, które umożliwi zdefiniowanie ustawień nowej gry. Użytkownik będzie mógł podać liczbę **n** oraz **k**, wskazać, kto wykonuje pierwszy ruch, a także wybrać kolory dla gracza i komputera. Poniższy rysunek przedstawia okno definiowania nowej gry:



Po wybraniu odpowiednich ustawień i kliknięciu przycisku **start** pojawi się główne okno gry:



Liczby możliwe do wybrania podczas gry reprezentowane będą przez przyciski, które będą zmieniać wygląd w zależności od podjętej przez graczy akcji.

Projekt interfejsu graficznego może ulec modyfikacjom.

Przebieg gry

Runda rozpoczyna się wybraniem przez jednego z graczy dwóch liczb. Po przyciśnięciu odpowiednich przycisków w głównym panelu, podświetlą się one na wyróżniony kolor, a wybrane liczby zostaną przypisane do przycisków w panelu „wybrane” po prawej stronie okna gry. Następnie drugi gracz wybierając jeden z tych dwóch przycisków wskazuje, którą liczbę koloruje na swój kolor. Odpowiednie przyciski w głównym panelu zmieniają wygląd zgodnie z wybranymi kolorami dla gracza i komputera. Pokolorowane przyciski stają się niemożliwe do wybrania przez graczy w kolejnych rundach. U góry i u dołu okna wyświetlane są wybrane już przez gracza i komputer liczby. Dolny panel zawiera informację o aktualnie rozgrywanej rundzie i ruchu, który aktualnie jest wykonywany (gracz/komputer wybiera dwie liczby, gracz/komputer koloruje liczbę).

Ruchy gracza-komputera przedstawiane będą poprzez symulowanie wybierania odpowiednich przycisków.

Architektura aplikacji

Projekt zostanie napisany w języku C#. Za pomocą technologii Windows Forms przygotowany zostanie interfejs graficzny.

Do przechowywania stanu gry stworzona będzie osobna klasa GameState:

```
class GameState
{
    List<int> availableNumbers;
    List<int> player;
    List<int> computer;
    int[] chosen;
    Movement currentMove;
}
```

Gdzie:

- **availableNumbers** to liczby dostępne do wybrania przez graczy
- **player** – liczby wybrane do tej pory przez gracza
- **computer** – liczby wybrane do tej pory przez komputer
- **chosen** – dwie liczby wybrane w danej rundzie przez jednego z graczy
- **currentMove** – typ ruchu obecnie wykonywanego przez jednego z graczy

Movement jest typem wyliczeniowym zdefiniowanym następująco:

```
public enum Movement
{
    PlayersChoice, // = 0
    ComputersChoice, // = 1
    PlayerColouring, // = 2
    ComputerColouring // = 3
}
```

Dla gracza-komputera przewidujemy dwie strategie gry (łatwą i trudną dla gracza).

Sprawdzanie wygranej

Sprawdzenie ewentualnej wygranej gracza lub komputera będzie mogło być zrealizowane w prosty sposób algorytmem siłowym (sprawdzenie różnic między wszystkimi liczbami ze zbioru).

Twierdzenia Szemeriediego wykorzystamy do sprawdzenia, czy na pewno gra zostanie skończona czyjąś wygraną, na podstawie podanych wartości liczb **N** i **k**. Odpowiedni komunikat zostanie wyświetlony użytkownikowi.

Dla danych wartości **N** i **k** gra zawsze skończy się czyjąś wygraną, jeśli najliczniejszy podzbiór niezawierający żadnego ciągu arytmetycznego o długości **k** ma licznosc mniejszą niż **N/2**.

Innymi słowy, dowolny podzbiór o licznosci **N/2** zawiera jakiś ciąg arytmetyczny długości **k**.

Czyli, korzystając z górnego oszacowania **N(k, d)**, stwierdzimy że gra zawsze skończy się wygraną, jeśli spełnione będzie:

$$N > 2^{2^d - 2^{k+9}}$$

Gdzie **d** wynosi nie wiem, ale może $N/2 - 1/N$

Strategie komputera

Strategia losowa

Komputer w swojej rundzie wybiera dwie liczby spośród jeszcze nie wybranych w sposób losowy. Mając do wyboru dwie liczby wybrane przez gracza, spośród nich wybiera jedną również w sposób losowy.

Strategia zachłanna

Komputer w swojej rundzie pierwszą liczbę wybiera taką, która wydłuży najdłuższy posiadany ciąg arytmetyczny. W przypadku kilku możliwości (wydłużanych ciągów lub możliwych liczb wydłużających ciąg) wybierana jest dowolna z rozważanych liczb. Dodatkowo, komputer sprawdza czy w taki sposób wydłużając ciąg, pozostaną wśród niewybranych liczby, które mogą posłużyć w kolejnych rundach do wydłużenia ciągu do długości **k**.

Druga liczba, a także jedna z dwóch liczb w przypadku rundy gracza, wybierana jest w taki sam sposób.

Przebieg prac

Projekt rozpoczniemy wykonaniem interfejsu. Strategia komputera zostanie zaimplementowana po uprzednim przygotowaniu matematycznego opisu problemu równoległe z implementacją sprawdzenia ewentualnej wygranej. Planowane terminy wykonania ww. prac to odpowiednio koniec listopada oraz koniec grudnia. Zostanie przygotowany również zestaw testów sprawdzających poprawność działania poszczególnych funkcjonalności aplikacji.