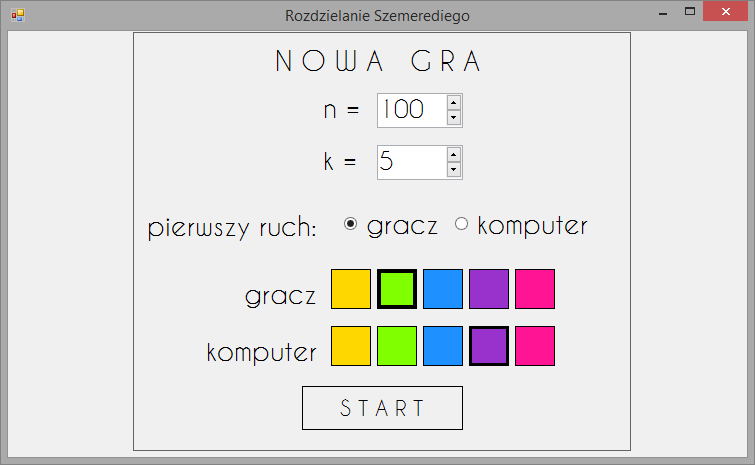
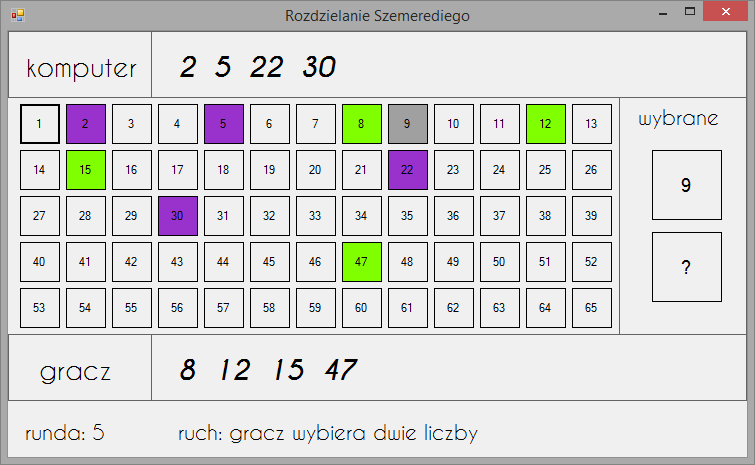
Projekt zostanie napisany w języku C#. Za pomocą technologii Windows Forms przygotowany zostanie interfejs graficzny.

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 wybranie kolorów dla gracza i komputera. Poniższy rysunek przedstawia okno definiowania nowej gry:



Po wybraniu odpowiednich ustawień 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.

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 zmienią 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.

Projekt interfejsu graficznego może ulec modyfikacjom.

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 w początkowej fazie projektu przewidujemy jedną strategię gry.

Interfejs i strategia komputera będą implementowane równolegle. Komponenty te mogą być rozwijane niezależnie od siebie, następnie połączone w całość.

- w jaki sposób będzie sprawdzana po każdym ruchu ewentualna wygrana? czy jest to trudny problem (jeśli tak, to w jaki sposób będą Państwo szukać sposobu jego rozwiązania) czy łatwy (jeśli tak, to jaka będzie ta metoda)?

- jaki jest Państwa plan działania? czy rozpoczną Państwo od interfejsu, a potem dołożą strategię komputera, czy może odwrotnie, czy równolegle? w jakich terminach zamierzają Państwo zakończyć poszczególne etapy pracy?

- w jaki sposób zamierzają Państwo sprawdzić, czy poszczególne części działają prawidłowo?