**Sprawozdanie z realizacji projektu ZPK – Gra Pong**

**1.** Informacje ogólne

* **Temat projektu**: Gra Pong – wersja komponentowa i niekomponentowa
* **Autorzy**: Serhii Yakubiv
* **Technologie**: C# (.NET), Windows Forms
* **Środowisko pracy**: Visual Studio 2022

**2.** Cel projektu

Celem projektu było stworzenie gry typu retro "Pong" w dwóch wersjach:

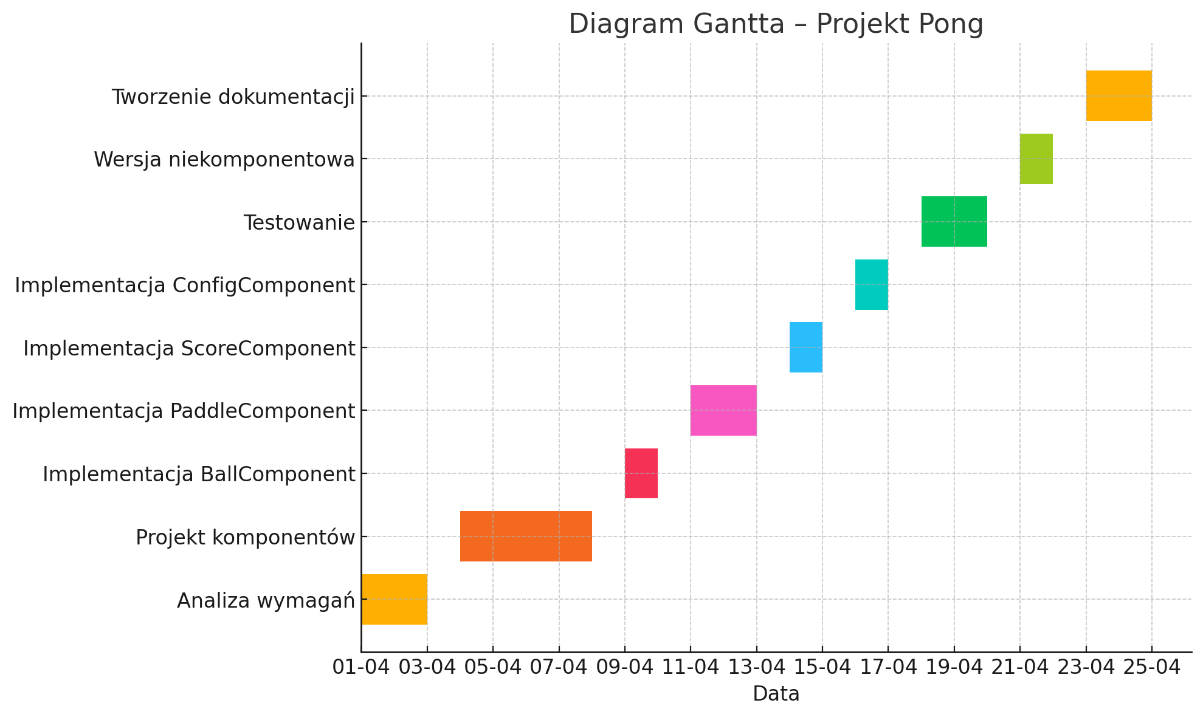
* **Wersja komponentowa**: oparta na zasadach programowania komponentowego, z wydzielonymi modułami odpowiadającymi za logikę piłki, paletek, systemu punktacji i konfiguracji.
* **Wersja niekomponentowa**: maksymalnie uproszczona, zrealizowana w jednym pliku C# bez podziału na komponenty.

Gra oferuje tryby:

* Gracz vs Gracz
* Gracz vs Komputer (AI)

Celem rozgrywki jest zdobycie ustalonej liczby punktów szybciej niż przeciwnik.

**3.** Diagram Gantta

**

**4.** Wymagania funkcjonalne

|  |  |
| --- | --- |
| **Wymaganie** | **Opis** |
| Ruch piłki | Piłka porusza się z ustaloną prędkością, odbija się od krawędzi i paletek |
| Ruch paletek | Paletki sterowane przez gracza lub AI |
| Kolizje | Obsługa odbić piłki i punktów |
| System punktacji | Zliczanie punktów, zakończenie gry po osiągnięciu limitu |
| Konfiguracja | Możliwość ustawienia nazw graczy, trybu gry, prędkości piłki |

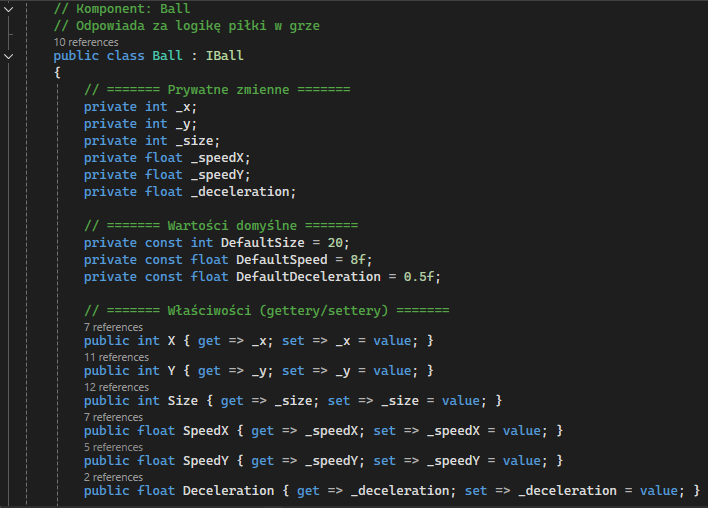
**5.** Wyodrębnione komponenty

|  |  |
| --- | --- |
| Komponent | Opis |
| BallComponent | Logika ruchu piłki i odbić |
| PaddleComponent | Logika paletek: sterowanie, AI |
| ScoreComponent | System punktacji i zapisu najlepszego wyniku |
| ConfigComponent | Konfiguracja gry, serializacja do pliku |

**6.** Charakterystyka komponentów

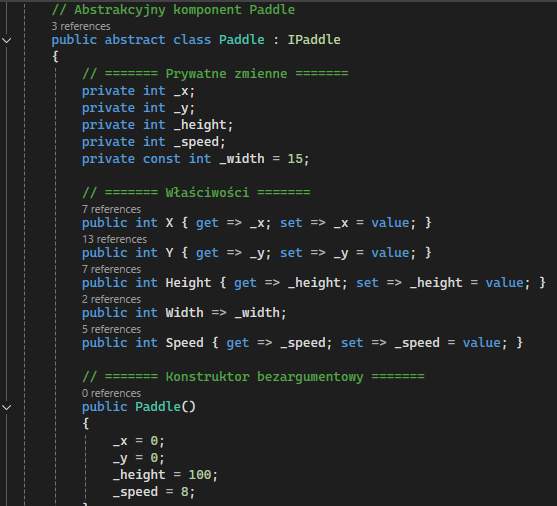
**BallComponent**

* **Zmiennych prywatnych**: 6
* **Default values**: rozmiar, prędkość, deakceleracja
* **Konstruktor bezargumentowy**: tak
* **Getter/Setter**: tak
* **Serializacja**: tak (JSON)
* **Testy**: BallTests.cs
* **Wywołanie**: ball.Move();, ball.Reset(800, 600);



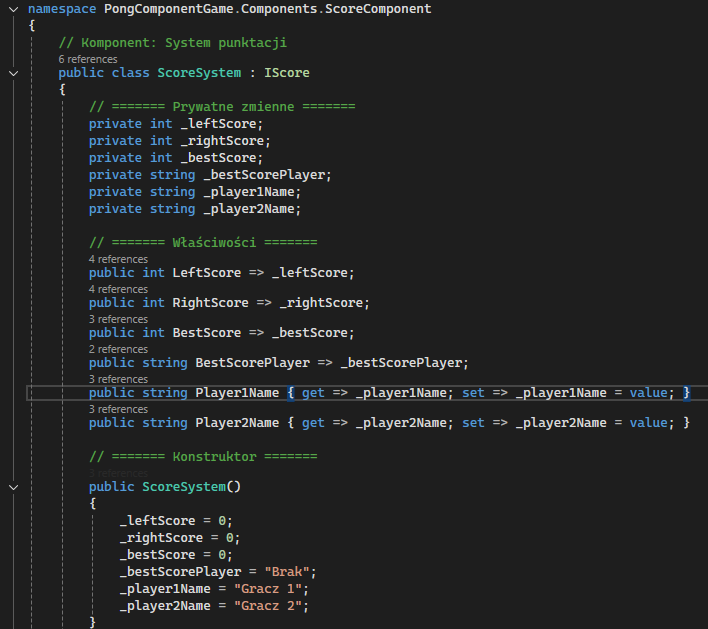
**PaddleComponent**

* Paletki dziedziczą po klasie abstrakcyjnej Paddle
* AI i gracz różnią się logiką sterowania
* Zmiennych prywatnych: 5+
* **Testy**: PaddleTests.cs



**ScoreComponent**

* Zliczanie punktów, najlepszy wynik
* **Zapis/odczyt JSON**: tak
* **Testy**: ScoreTests.cs

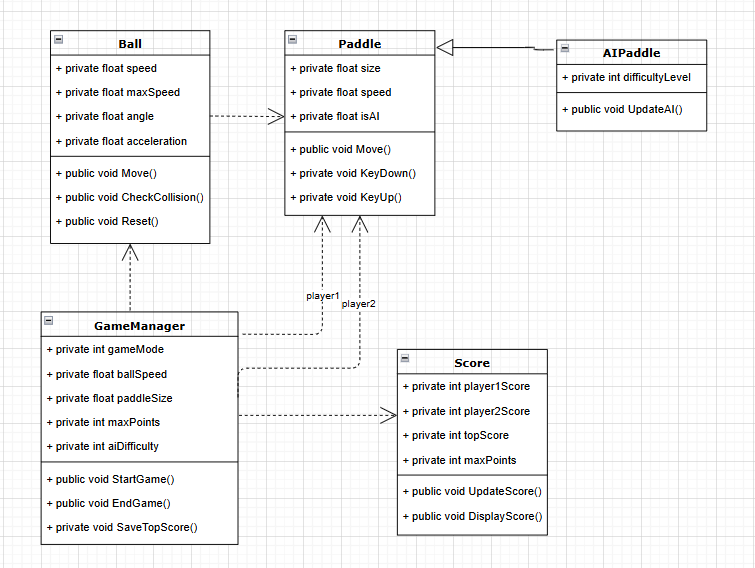


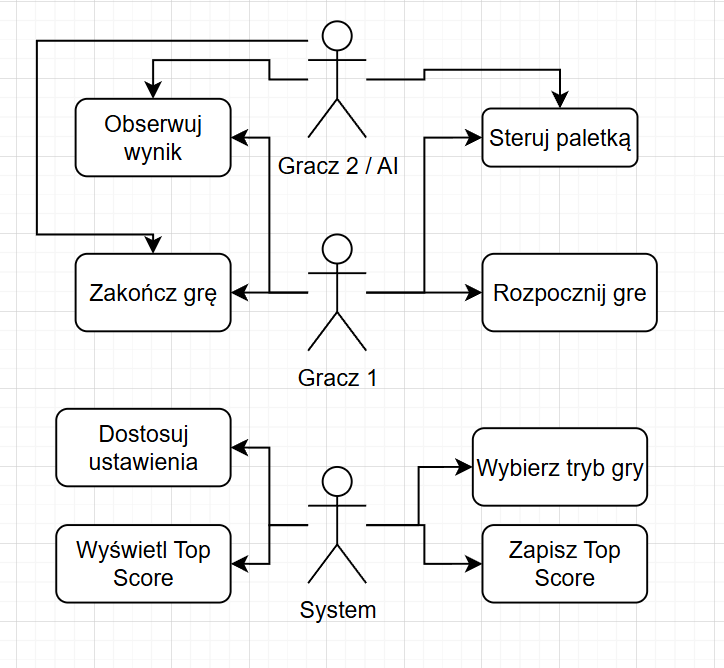
**ConfigComponent**

* Przechowuje ustawienia gry
* Możliwość zmiany przez formularz
* **Testy**: ConfigTests.cs



**7.** Diagramy klas i przypadków użycia





**8.** Wersja niekomponentowa (C#)

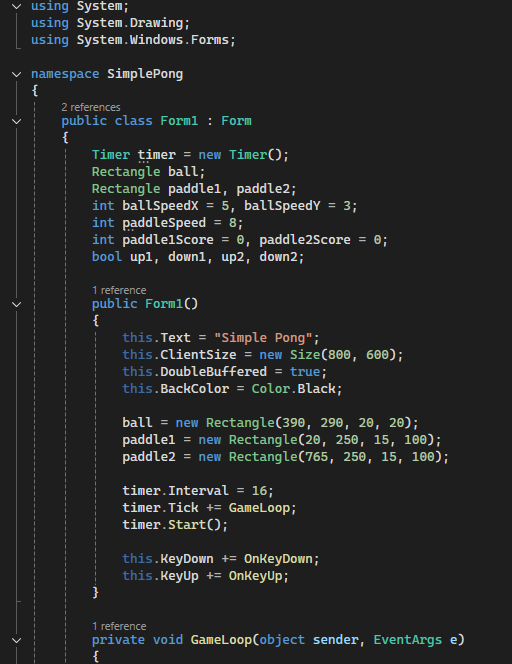
* Implementacja w jednym pliku: SimplePong.cs
* Brak podziału na klasy
* Wszystkie zmienne i logika (piłka, paletki, punktacja) zdefiniowane lokalnie
* Obsługa gry przez Form1, Timer i zdarzenia klawiatury

**Zalety**:

* Prostota
* Szybka implementacja

**Wady**:

* Brak modularności
* Trudna rozbudowa



**9. Porównanie wersji komponentowej i niekomponentowej**

|  |  |  |
| --- | --- | --- |
| Cecha | Komponentowa | Niekomponentowa |
| Modularność | ✅ | ❌ |
| Czytelność kodu | ✅ | ❌ |
| Rozszerzalność | ✅ | ❌ |
| Liczba plików | wiele | 1 |
| Testy jednostkowe | ✅ | ❌ |

**10. Testowanie**

* Debug.Assert w testach komponentów
* RunAll() w testach Ball, Paddle, Score, Config

**11. Wnioski końcowe**

* Projekt spełnia wszystkie podstawowe wymagania ZPK
* Gra jest w pełni funkcjonalna
* Dzięki modularnej strukturze można łatwo rozwijać projekt
* Wersja uproszczona (SimplePong.cs) pokazuje kontrast i potwierdza zalety komponentów

**12. Zestawienie użytych narzędzi**

* Visual Studio 2022
* .NET 6 / Windows Forms
* JSON (serializacja)

**13. Źródła / materiały pomocnicze**

* Dokumentacja Microsoft C#
* Przykłady gier Pong online