

Zastosowanie programowania komponentowego

Michał Golik

Adrian Łoński

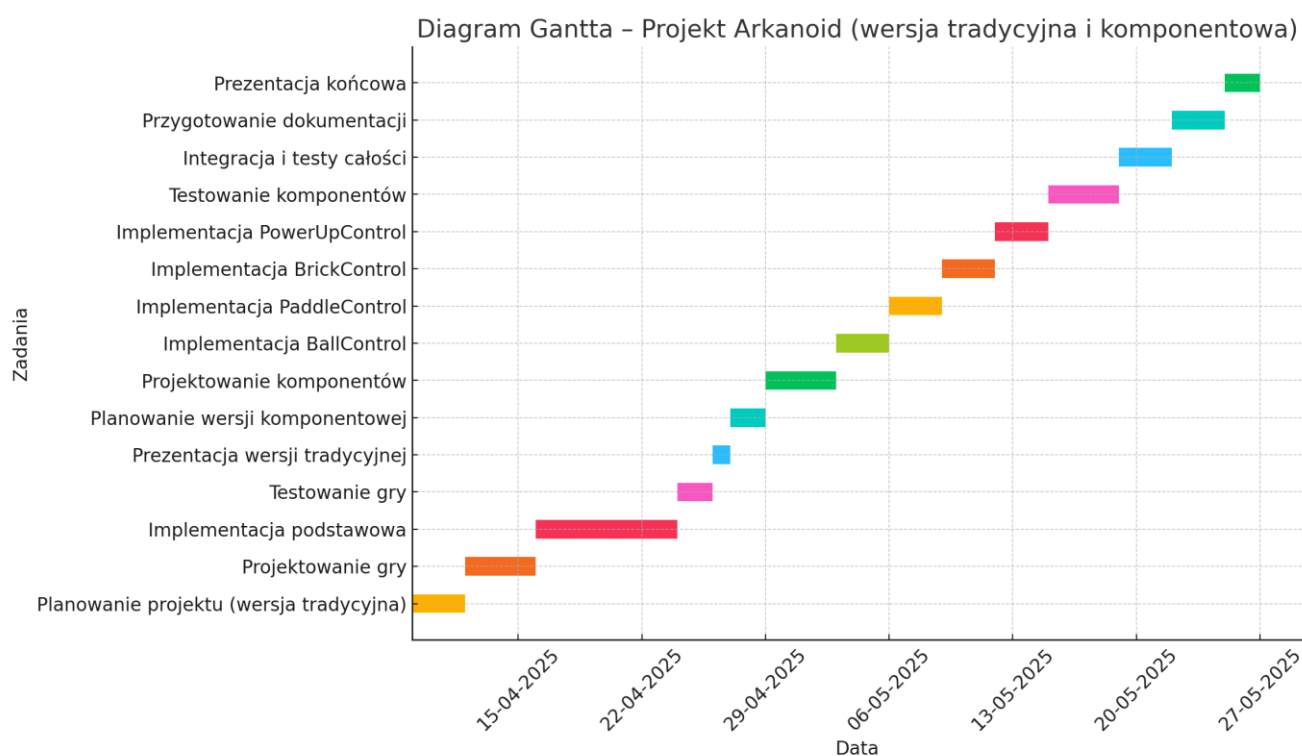
Politechnika koszalińska

20.05.2025

1. Temat projektu

Gra Arkanoid - klasyczna gra zręcznościowa typu "**breakout**", w której gracz steruje **paletką** (platformą) u dołu ekranu i odbija **piłkę**, aby zniszczyć wszystkie **cegiełki** znajdujące się na górze planszy.

2. Diagram Gantta



3. Opis projektu oraz cel

3.1 Etap 1 – Gra Arkanoid (wersja tradycyjna)

- **Opis:** Prosta gra typu Arkanoid z podstawową strukturą, bez podziału na komponenty.
- **Moduły:** Jeden główny plik z logiką gry.
- **Wyzwania:** Trudniejsza rozbudowa, brak separacji logiki.
- **Efekt:** Działająca gra, użyta jako baza do rozwoju wersji komponentowej.

3. Etap 2 – Gra Arkanoid (wersja komponentowa)

- **Opis:** Refaktoryzacja gry do postaci komponentowej.
- **Komponenty:**
 - BallControl – kontroluje ruch piłki i kolizje
 - BrickControl – zarządza cegiełkami
 - PaddleControl – steruje paletką gracza
 - PowerUpControl – zarządza bonusami
- **Zalety:** Łatwiejsze testowanie, lepsza skalowalność, zgodność z zasadami SOLID.
- **Dokumentacja:** Każdy komponent posiada opis, metody, typy zmiennych i konstruktor.

4. Lista wymagań funkcjonalnych

- Start gry
- Odbijanie piłki paletką
- Niszczanie cegiełek
- Zbieranie bonusów
- Zliczanie punktów
- Wyświetlanie rankingu
- Obsługa przegranej i wygranej

5. Komponenty:

5.1 BallControll - odpowiada za ruch piłki oraz kolizje z innymi obiektami.

- Kontrolka GUI odpowiedzialna za ruch kulki
- Zawiera własne pola stanu (`VelocityX`, `VelocityY`)
- Udostępnia metody do poruszania się i odbijania
- Działa niezależnie, może być wielokrotnie używana w różnych projektach

5.2 BrickControll - odpowiada za zarządzanie cegiełkami i ich zniszczenie.

- Reprezentuje pojedynczy klocek (element GUI)
- Inkapsuluje swój stan (`HitPoints`), logikę uszkodzeń
- Emituje zdarzenie `BrickDestroyed` (komunikacja z otoczeniem)
- Można go tworzyć, usuwać, nasłuchiwać zdarzeń niezależnie

5.3 PaddleControl - odpowiada za ruch paletki sterowanej przez gracza.

- Samodzielna kontrolka GUI (dziedziczy `UserControl`)

- Inkapsuluje logikę ruchu paletki (metody MoveLeft, MoveRight)
- Odpowiedzialny za swój wygląd i zachowanie
- Można ją używać i konfigurować niezależnie od reszty aplikacji

5.4 PowerUpControl - odpowiada za generowanie i obsługę bonusów.

- Reprezentuje bonus jako samodzielną kontrolkę
- Posiada właściwość Type definiującą zachowanie
- Zawiera animację spadania (Timer), obsługę zdarzenia Collected
- Może być używany, skonfigurowany i włączany do gry niezależnie

6. Testy

Przypadki testowe:

PaddleControl

Cel: sprawdzić, czy paletka się pojawia i reaguje na strzałki.

- **Kroki:**

1. Uruchom grę (F5).
2. Upewnij się, że po max. 10 sekundach na dole ekranu widać prostokątną paletkę.
3. Kliknij tło, naciśnij → i ← sprawdź czy reaguje.

- **Oczekiwany wynik:**

1. Paletka jest widoczna zaraz po starcie.
2. Po naciśnięciu strzałek przesuwa się w lewo i w prawo.

- **Status testu:**

1. Zdane jeśli paletka widać i porusza się.
2. Nie zdane jeśli nie widać paletki lub nie reaguje na klawisze.

BallControl

Cel: sprawdzić rysowanie i odbicia kulki.

- **Kroki:**

1. Uruchom grę.
2. Obserwuj kilka sekund ruchu kulki (powinna automatycznie ruszyć po starcie).

3. Sprawdź, czy kulka odbija się od lewej, prawej i górnej krawędzi okna.

- **Oczekiwany wynik:**

1. Kulka jest widoczna jako czerwony okrąg.
2. Odbija się od bocznych i górnej krawędzi bez “przechodzenia” przez ramkę.

- **Status testu:**

1. Zdane jeśli kulka widać i zachowuje się poprawnie.
2. Nie Zdane jeśli nie widać kulki lub nie odbija się prawidłowo.

BrickControl

Cel: sprawdzić pojawienie się i usuwanie cegiełek.

- **Kroki:**

1. Uruchom grę.
2. Sprawdź, czy na górze ekranu jest rząd cegiełek (albo wzór rombu/“X” na poziomach 2 i 3).

3. Uderz kulką w kilka cegiełek i obserwuj, czy znikają po trafieniu.

• Oczekiwany wynik:

1. Cegielki są równo rozmieszczone według aktualnego poziomu.

2. Po trafieniu znikają.

• Status testu:

1. Zdane jeśli cegielki się pojawiają i znikają przy kolizji.

2. Nie Zdane jeśli nie ma cegiełek lub nie reagują na uderzenia.

PowerUpControl

Cel: sprawdzić spadanie i zbieranie bonusów.

• Kroki:

1. uruchom grę.

2. Traf cegielkę, żeby wygenerować bonus.

3. Obserwuj złotą ikonkę spadającą w dół.

4. Zbierz ją paletką (kolizja).

• Oczekiwany wynik:

1. Bonus pojawia się w miejscu ciosu i spada w dół.

2. Po zetknięciu z paletką bonus znika i jego efekt (np. powiększenie paletki lub dodatkowe życie) zostaje zastosowany.

• Status testu:

1. Zdane jeśli bonus widać, spada i reaguje na kolizję.

2. Nie Zdane jeśli nie generuje się, nie spada lub nie daje efektu.

Liczba żyć (1–10)

1. Kroki

- **Uruchom aplikację i w Menu wybierz Opcje.**
- **W polu Liczba żyć spróbuj ustawić wartość 0**
- **Następnie spróbuj ustawić wartość 11**

2. Oczekiwany wynik

- **Minimalna wartość nie może spaść poniżej 1.**
- **Maksymalna wartość nie może przekroczyć 10.**

3. Status

- **Zdane jeśli po próbie wpisania 0 widzisz 1, a przy próbie wpisania 11 widzisz 10.**
- **Nie Zdane jeśli kontrolka pozwala na wartości spoza zakresu 1–10.**

Punkty za klocek (500–1000)

1. Kroki

- **Nadal w oknie Opcje, w polu Punkty za klocek spróbuj wpisać wartość 400 (poniżej 500).**
- **Następnie spróbuj wpisać 1200 (powyżej 1000).**

2. Oczekiwany wynik

- **Minimalna dozwolona wartość to 500. Po wpisaniu 400 kontrolka powinna skorygować na 500.**
- **Maksymalna dozwolona wartość to 1000. Po wpisaniu 1200 kontrolka powinna skorygować na 1000.**

3. Status

- **Zdane jeśli przy próbie wpisania 400 widzisz 500, a przy 1200 widzisz 1000.**
- **Nie Zdane jeśli kontrolka przyjmuje wartości spoza zakresu 500–1000.**

7. Porównanie podejść

Cecha	Wersja tradycyjna	Wersja komponentowa
Struktura kodu	Monolityczna	Modularna (komponenty)
Łatwość rozbudowy	Niska	Wysoka
Możliwość testowania	Ograniczona	Wysoka (testy jednostkowe)
Czytelność kodu	Średnia	Dobra
Obsługa wyjątków	Minimalna	Ulepszona

8. Zestawienie użytych narzędzi programistycznych

- Visual Studio Code 2022
- .NET
- C#

9. materiały źródłowe

Repozytorium stworzono na GitHub:

https://github.com/Michael4254/ZPK_Lato2025_Golik_Lonski

10. Wnioski

Projekt został zrealizowany. Gra działa zgodnie z założeniami.
Można rozbudować ją o kolejne poziomy i nowe bonusy.