

# Pac-Man Arena

---

## Cel projektu

Celem projektu jest rozwijanie umiejętności programowania w C++ (szczególnie programowania obiektowego) oraz zapoznanie się z zagadnieniami tworzenia gier przy użyciu biblioteki SFML. Projekt zakłada implementację gry Pac-Man w trybie wieloosobowym (serwer-klient), z podziałem na widoki gry oraz prostą synchronizacją stanu rozgrywki.

## Rejestr decyzji

- **Grafika i rendering:** Wykorzystano SFML do rysowania kształtów na ekranie (zamiast gotowych tekstur), co pozwala skupić się na logice gry i obsłudze obiektów.
- **Struktura aplikacji:** Projekt został podzielony na aplikację serwerową i kliencką.
  - **Serwer:** Odpowiada za utrzymywanie połączeń oraz komunikację między klientami.
  - **Klient:** Uruchamia interfejs graficzny, przetwarza dane wejściowe i synchronizuje stan gry.

## Status projektu

Co działa:

- **Logika mapy i inicjalizacja encji:**
  - Mapa gry jest zdefiniowana jako tablica dwuwymiarowa, gdzie poszczególne wartości odpowiadają ścianom, kropkom, power dotom, pozycji startowej Pac-Mana oraz duchów (*game.cpp*, *game.h*).
  - Metoda `set_entities_positions()` poprawnie tworzy i umieszcza obiekty `Dot`, `PowerDot`, `Pacman` oraz `Ghost` w odpowiednich miejscach na mapie.
- **Rysowanie i rendering:**
  - Klasa `Drawer` odpowiedzialnie renderuje planszę, obiekty gry oraz wynik rozgrywki (*drawer.cpp*, *drawer.h*).
  - Zastosowano hierarchię rysowania: najpierw kropki, potem duchy, a na końcu Pac-Man.
- **Ruch i logika obiektów:**
  - `Pacman` porusza się zgodnie z wprowadzonymi kierunkami, korzystając z zaimplementowanej metody `update()` (*pacman.cpp*).
  - `Ghost` porusza się losowo oraz zmienia kierunek, gdy napotka przeszkodę (*ghost.cpp*).
- **Obsługa klienta i serwera:**
  - Klasa `Client` nawiązuje (lub symuluje) połączenie z serwerem oraz inicjuje rozgrywkę, dzieląc widok na 4 części (*client.cpp*, *client.h*).
  - Klasa `Server` nasłuchuje na określonym porcie, akceptuje połączenia i wysyła komunikat powitalny do klientów (*server.cpp*, *server.h*).

Czego brakuje:

### 1. Synchronizacja sieciowa:

- Aktualnie stan gry na kliencie jest niezależny – synchronizacja między wieloma klientami przez serwer w celu odtworzenia wspólnej rozgrywki nie została jeszcze zaimplementowana.

- Rozbudowa trybu lobby: potwierdzanie gotowości graczy, czekanie na czterech uczestników, zarządzanie ponownym dołączaniem do gry.
- 2. **Obsługa kolizji z duchami:**
  - W grze obecnie wykrywane są kolizje tylko między Pac-Manem a kropkami/power dotami, a interakcja z duchami (np. utrata życia, zakończenie gry) wymaga implementacji.
- 3. **Rozszerzenie logiki gry i interfejsu użytkownika:**
  - Implementacja pełnego menu (start, pomoc, wynik końcowy) oraz opcji wyjścia z gry za pomocą przycisków.
- 4. **Optymalizacja i refaktoryzacja:**
  - Rozbudowa algorytmu ruchu duchów – obecnie losowy, a w przyszłości możliwe wdrożenie algorytmu A\* dla inteligentniejszego ruchu.
- 5. **Dźwięk**
  - SFML obsługuje też dźwięk...

Link do repozytorium

<https://github.com/dunajski/PacManArena>

Autor

Michał Dunajski – [michal.dunajski.stud@pw.edu.pl](mailto:michal.dunajski.stud@pw.edu.pl)