

“BIT OF PIZZA” - INFORMACJE OGÓLNE

Nazwa projektu: Bit of Pizza

Opis: Bit of Pizza to prosta gra napisana w języku C++, w której gracz wciela się w pizzermana. Zadaniem gracza jest kompletowanie zamówień klientów poprzez wybór odpowiednich składników, pieczenie pizzy i dostarczanie jej na czas. Projekt kładzie nacisk na zarządzanie czasem i szybkość reakcji. Wszystkiemu towarzyszy humorystyczna piosenka w tle.

Główne funkcje:

- Obsługa zamówień klientów.
- Dynamiczne tworzenie pizz z dostępnych składników.
- System punktacji na podstawie dokładności i czasu realizacji zamówień.
- Oprawa dźwiękowa i graficzna

Technologie użyte w projekcie:

- Język programowania: C++ (standard C++11 lub nowszy)
- Silnik graficzny i dźwiękowy: Raylib
- Własne lub generowane grafiki (część wykonana ręcznie, część z pomocą AI)

Autorzy: Oliwia Sozańska i Blanka Podziemska

Data rozpoczęcia projektu: 09.04.2025

Status projektu: Skończona gra

WYMAGANIA SYSTEMOWE I ŚRODOWISKOWE

Wersja kompilatora C++:

- Minimalna wersja języka C++: C++11

Zalecane kompilatory:

- GCC 11 lub nowszy (Linux)
- MSVC 2019 lub nowszy (Windows)
- Clang 13 lub nowszy (Linux/macOS)

! Wymagana jest obsługa standardu C++11 ze względu na składnię i funkcje językowe użyte w kodzie !

System operacyjny:

- Windows (rekomendowany, testowany)
- Linux (kompilacja możliwa, ale może wymagać ręcznej konfiguracji ścieżek do raylib i grafik)
- macOS (możliwa kompilacja, ale nie testowana)

Zewnętrzne biblioteki:

- Raylib (prosta biblioteka C/C++ do tworzenia gier 2D/3D)

Wersja zalecana: 4.5 lub nowsza

Wymagania sprzętowe:

Gra jest lekka i nie wymaga dedykowanej karty graficznej ani wysokiej wydajności – idealna do uruchamiania na starszym sprzęcie.

INSTALACJA

1. Upewnij się, że masz zainstalowany:

- Kompilator **g++** (np. MinGW-w64)
 - Bibliotekę **raylib** (zainstalowaną i poprawnie skonfigurowaną)
2. **Otwórz terminal w katalogu z plikami źródłowymi** (tam, gdzie są main.cpp, gra.cpp, muzyka.cpp, klient.cpp).
 3. **Wpisz komendę:**


```
g++ main.cpp gra.cpp muzyka.cpp klient.cpp -o apka.exe -lraylib -lopengl32 -lgdi32 -lwinmm
```
 4. **Po udanej kompilacji uruchom aplikację wpisując w terminalu**


```
./apka.exe
```

2 SPOSÓB

1. Zrób krok pierwszy z 1 sposobu
2. Zainstaluj BitOfPizza.zip
3. Otwórz folder, kliknij na skrót i gotowe!

! Upewnij się, że wszystkie pliki są w jednym folderze !

STRUKTURA PROJEKTU

Struktura plików:

- **Pliki główne:**

- **main.cpp** – punkt wejścia do programu, inicjalizuje grę i rozpoczyna główną pętlę gry. W niej jest menu (4 różne przyciski), przygotowanie pizzy, działanie pieca oraz profil użytkownika.

- **Pliki wspierające:**

- **gra.cpp / gra.hpp** – obsługa postaci

- **klient.cpp / klient.hpp** – obsługa logiki klientów, ich zamówień i zachowań. W tych plikach ustawiony jest czas, sprawdzana jest poprawność zamówienia oraz czy gra jest skończona

- **zamowienia.cpp / zamowienia.hpp** – obsługa zamówień klientów. Stworzone są tu rodzaje pizz oraz dodawanie/odejmowanie punktów zależnie od zadowolenia klienta, przypieczenia oraz czasu.

- **muzyka.cpp / muzyka.hpp** – obsługa muzykę: aktualizowanie jej, włączanie i wyłączanie.

- **Zasoby:**

- **obrazki/** – katalog zawierający grafiki używane w grze, takie jak tła, postacie, składniki pizzy itp.
- **muzyka/** – katalog z plikami dźwiękowymi, w tym muzyką tła.

- **Inne pliki:**

- BIT OF PIZZA.zip – archiwum zawierające pełną wersję gry lub dodatkowe zasoby
- .gitignore – plik konfiguracyjny Git.
- README.md – plik z opisem projektu, instrukcjami instalacji i uruchomienia gry.

DOKŁADNY OPIS

Opis ogólny:

Plik main.cpp jest centralnym punktem uruchomieniowym gry "Bit of Pizza". Odpowiada za inicjalizację okna gry, ładowanie zasobów (tekstur, muzyki), sterowanie ekranami oraz zarządzanie logiką głównego przebiegu gry.

Funkcjonalności:

- Inicjalizacja okna (800x600 px).
- Wczytanie zasobów graficznych: tekstury tła, składników pizzy, pizzy, interfejsów.
- Wczytanie muzyki za pomocą klasy **Muzyka**.
- Obsługa stanów gry (ekrany: menu, gra, przygotowywanie pizzy, piec, profil, instrukcja, koniec gry).
- Obsługa interakcji gracza z interfejsem oraz przełączanie ekranów za pomocą myszki.
- Zastosowanie logiki przygotowywania pizzy oraz pieczenia pizzy.
- Współpraca z klasą **Klienci**, która zarządza systemem zamówień i ocenianiem gracza.

Używane klasy i moduły:

- gra.hpp – logika ekranu gry.
- klienci.hpp – obsługa klientów, ich zamówień i oceniania gracza.
- muzyka.hpp – odtwarzanie muzyki i aktualizacja stanu.
- zamowienia.hpp – system zarządzania zamówieniami.

2. Plik: gra.hpp

Opis:

Zawiera funkcję **UruchomGre(Texture2D)**, która wyświetla tło, stawia meble na poprawne miejsce, ładuje kucharza, obsługuje sterowanie graczem, blokuje graczowi poruszanie się po meblach.

3. Plik: klient.hpp

Opis:

Zawiera klasę z metodami:

- rysuj() – rysowanie klienta, lub tekstu związanego z zamówieniem.
- aktualizuj() – aktualizacja czasu, zmiana klientów.
- sprawdzInterakcje() – czy zamówienie zostało przyjęte, jeśli tak i gracz ponownie kliknie na klienta, sprawdza poprawność przygotowanej pizzy (składniki, stan wypieczenia) - poprzez wywołanie funkcji **sprawdzPoprawnosc()**.
- getPunkty(), getZrobionePizze(), getZadowoleniKlienci() – zwracają wartości.
- czyGraSkonczone() – zwraca informację o końcu gry (po minięciu czasu na zamówienie, gra się kończy).
- pozostalyCzas() – ile czasu zostało na wykonanie aktualnego zadania.

4. Plik: muzyka.hpp

Opis:

Klasa Muzyka, konstruktor ładuje plik z muzyką tła.

- Aktualizuj() – cykliczne aktualizowanie stanu odtwarzania.
- Zakonczenie() – „zamyka” muzykę.

8. Plik: zamowienia.hpp

Opis:

Zarządzanie zamówieniami klientów:

- wylosujPizzaID() – losuje id pizzy.
- sprawdzPoprawnosc() – sprawdza, czy pizza jest dobrze wykonana.
- tekstZamowienia() – zwraca nazwę zamówionej pizzy.
- getCzyZadowolony(), getPunkty(), getPizzaID(), getZrobionePizze(), getZadowoleniKlienci() – zwraca wartości.