

# Implementacja projektu

Zrealizowany program udostępnia funkcjonalność wirtualnej kamery oraz eliminację elementów zasłoniętych za pomocą algorytmu malarskiego.

## Szczegóły techniczne

Program został zaimplementowany w języku Python (wersja 3) z użyciem następujących bibliotek:

- [Pygame](#) – biblioteka do pisania prostych gier. Wykorzystano jedynie funkcje służące do:
  - Rysowania wielokątów na ekranie.
  - Rysowania linii na ekranie.
  - Obsługi klawiatury.
- [Numpy](#) – biblioteka do obliczeń matematycznych.

## Założenia

- Na wirtualnej scenie znajdują się prostopadłościany imitujące budynki.
- Obiekty na scenie nie nachodzą na siebie.
- Wchodzenie do “budynków” nie zostało zaimplementowane.
- Kamera znajduje się zawsze pośrodku układu współrzędnych.
- Operacje kamery działają na całej scenie.
- Przycinanie (*clipping*) nie zostało zaimplementowane.
- Ekran ma inny układ współrzędnych — punkt (0, 0) znajduje się w lewym, górnym rogu ekranu.

## Sterowanie

Użytkownik może sterować kamerą za pomocą następujących klawiszy:

- ↑ — ruch do przodu (wzrastające współrzędne osi Z);
- ↓ — ruch do tyłu (malejące współrzędne osi Z);
- ← — ruch w lewo (malejące współrzędne osi X);
- → — ruch w prawo (rosnące współrzędne osi X);
- d — ruch do góry (rosnące współrzędne osi Y);
- c — ruch w dół (malejące współrzędne osi Y);
- z — powiększenie (zoom in);
- x — zmniejszenie (zoom out);
- r — powrót kamery do położenia początkowego (reset);
- a — obrót w lewo (dookoła osi Y);
- s — obrót w prawo (dookoła osi Y);

- f — obrót do góry (dookoła osi X);
- v — obrót w dół (dookoła osi X);
- g — pochylenie w lewo (dookoła osi Z);
- b — pochylenie w prawo (dookoła osi Z);
- p — zmiana trybu rysowania:
  - rysowanie krawędzi prostokątów,
  - rysowanie według algorytmu malarskiego;
- q — wyjście z programu (równoważne naciśnięciu przycisku x);
- += — zwiększenie szybkości poruszania
- -\_ — zmniejszenie szybkości poruszania

Prezentacja działania programu znajduje się w filmiku: `virtual-camera.mp4`, w którym zostały przedstawione wszystkie wyżej wymienione operacje.

## Instalacja

Wymagany jest manager pakietów dla Pythona — [pip](#). Jest on domyślnie zainstalowany, jeżeli Python został pobrany z [oficjalnej strony](#). Następnie należy zainstalować wspomniane wcześniej biblioteki za pomocą następującej komendy w linii poleceń `pip install -r requirements.txt`. Istnieje możliwość ręcznego zainstalowania bibliotek za pomocą dwóch poleceń:

- `pip install pygame`
- `pip install numpy`

## Uruchamianie

Program można uruchomić na dwa sposoby:

- Uruchomienie pliku wykonywalnego, który można zbudować za pomocą biblioteki `pyinstaller` lub `py2exe`.
- Wpisanie w linii poleceń `python camera.py`.

## Podsumowanie

W projekcie udało się zaimplementować prostą scenę trójwymiarową wraz z algorytmem eliminacji elementów zasłoniętych. Podczas wybierania biblioteki do wykonywania rysowania 2D na ekranie zauważyłam, że praktycznie wszystkie mają wbudowane operacje na scenie trójwymiarowej. Od programisty oczekują one zbudowania sceny, a silnik graficzny jest gotowy. Jeżeli chodzi o rozmaite optymalizacje, standardem jest na przykład *frustum culling*, który jest bardziej zaawansowany niż algorytm malarski. Darmowe biblioteki takie jak `pygame` są bardzo popularne i są używane nie tylko w amatorskich projektach, lecz także w komercyjnych grach, które można zakupić na platformie Steam.