

# Przewijana gra akcji 2D w języku JavaScript z wykorzystaniem WebGL

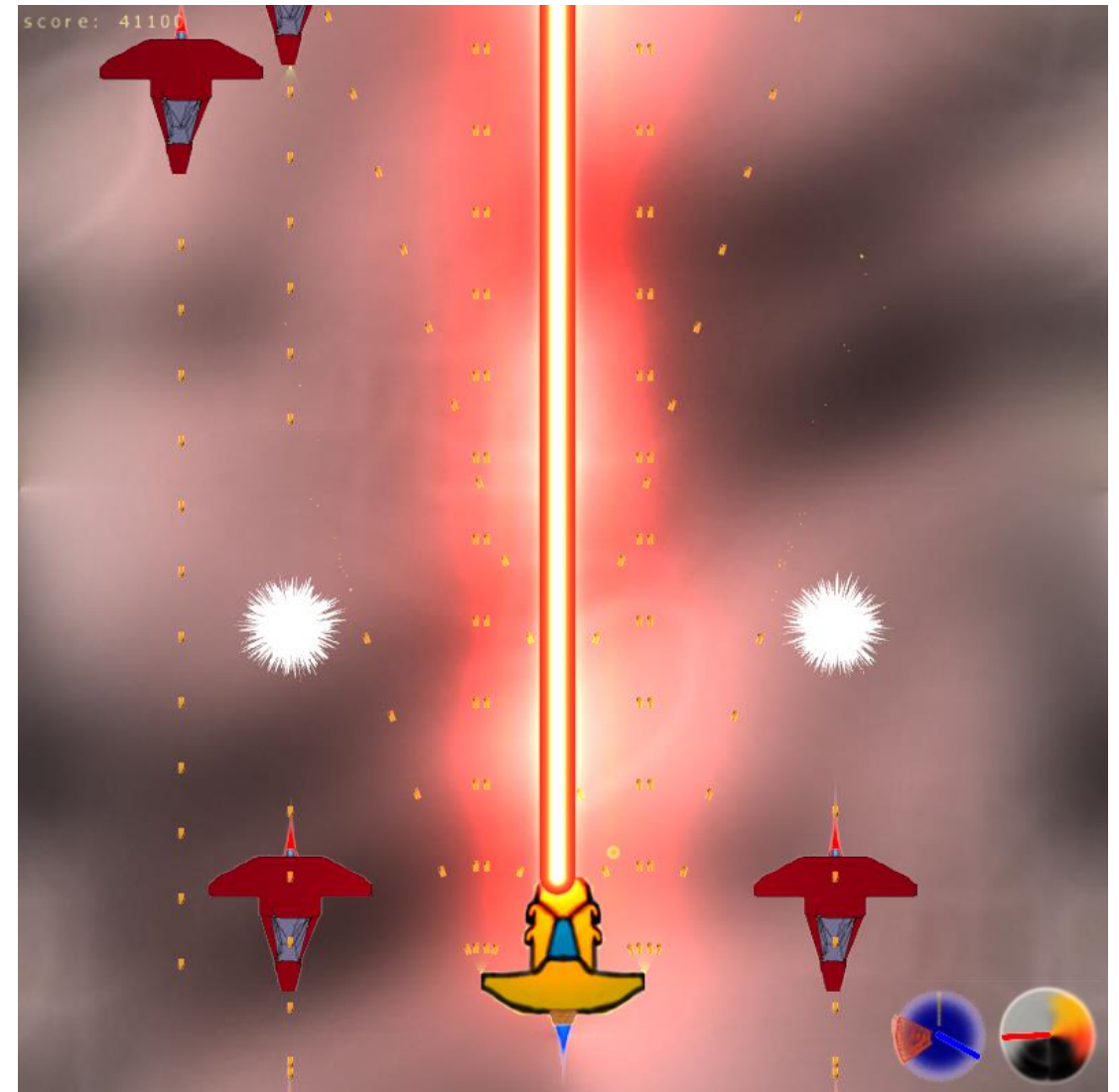
Szymon Lewandowski

# Plan prezentacji:

- Zasady gry
- Główne części gry
- Grafika:
  - Vertex shader
  - Pixel shader
  - Przykładowe shadery
- System kolizji:
  - Broad phase
  - Narrow phase
- Dźwięk:
  - Efekty dźwiękowe
  - Muzyka

# Ogólne zasady gry

- Menu
- Poziomy
- Obiekty
- Statek gracza
- Przeciwnicy
- Punkty wytrzymałości i przegrzanie
- Broń i pociski
- Punkty i pieniądze



# Najważniejsze części w grze

- Logika gry
  - Aktualizacja wszystkich obiektów w grze
  - Interakcje między obiektami
  - Aktualizacja wszystkich obiektów po interakcjach
- Grafika
  - Shadery
  - Tekstury i modele
- Muzyka
  - Efekty dźwiękowe
  - Muzyka

# Logika gry

- Główna pętla gry
- Struktura drzewa
- Aktualizacje obiektów
- Zderzenia
- Wynik zderzenia
- Inne interakcje między obiektami

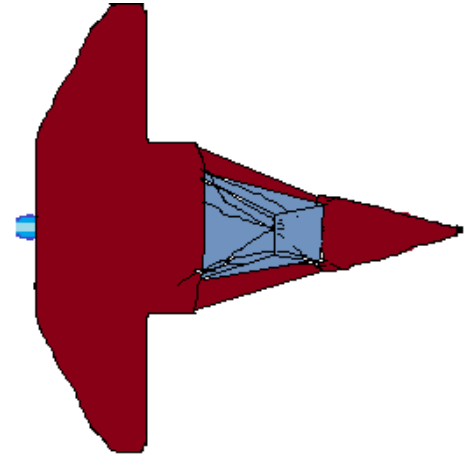
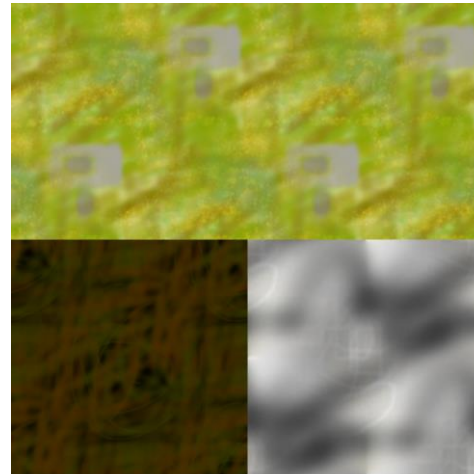
# Grafika

## Shader:

- Program działający na karcie graficznej
- Vertex shader
- Pixel (fragment) shader

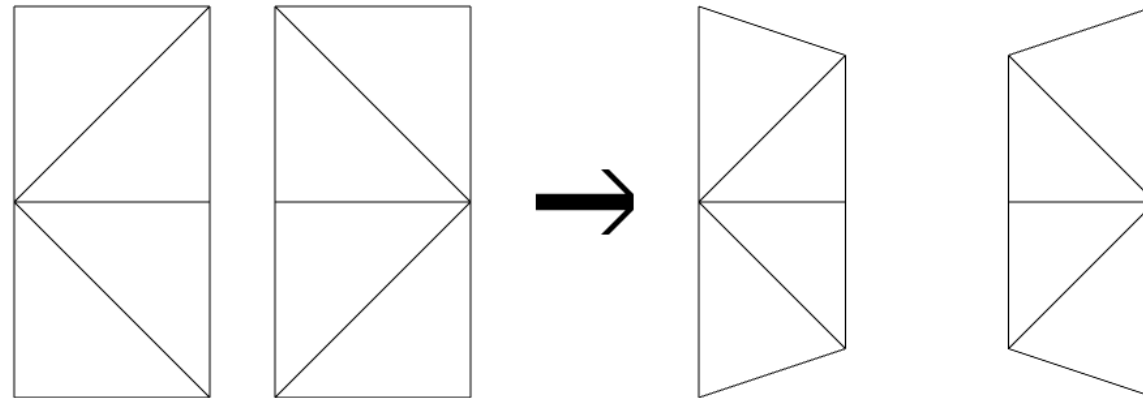
## Część artystyczna:

- Modele
- Tekstury
- Efekty



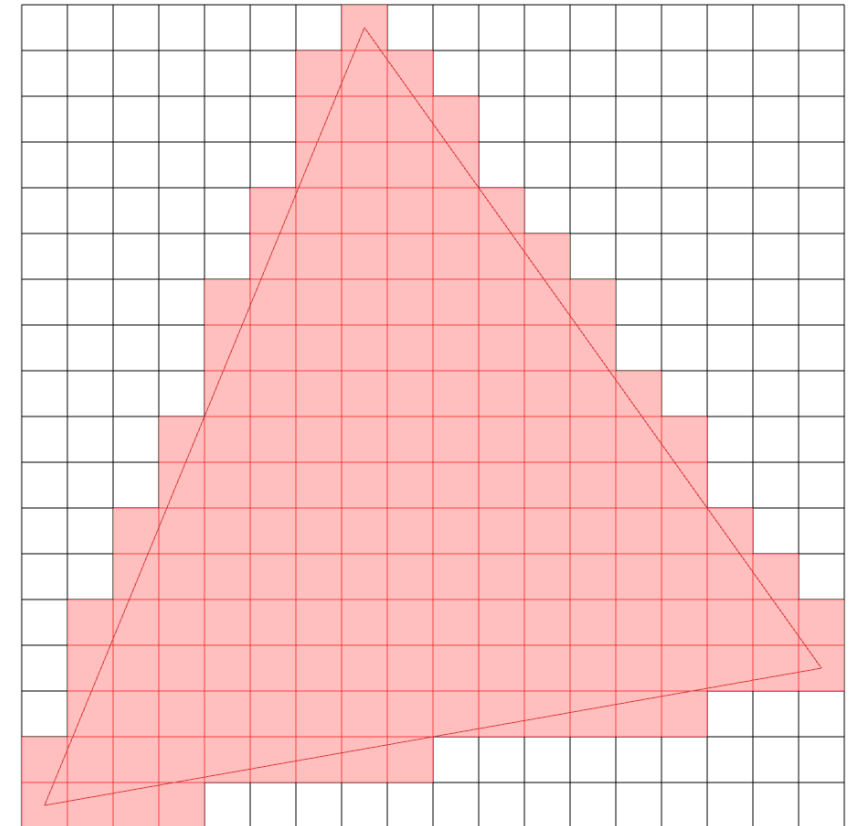
# Vertex Shader

- Przetwarza wierzchołki
- Zazwyczaj zajmuje się przetwarzaniem pozycji wierzchołków przez macierz obrotu, perspektywy, przesunięcia itp.
- Przekazuje dane do rasteryzatora



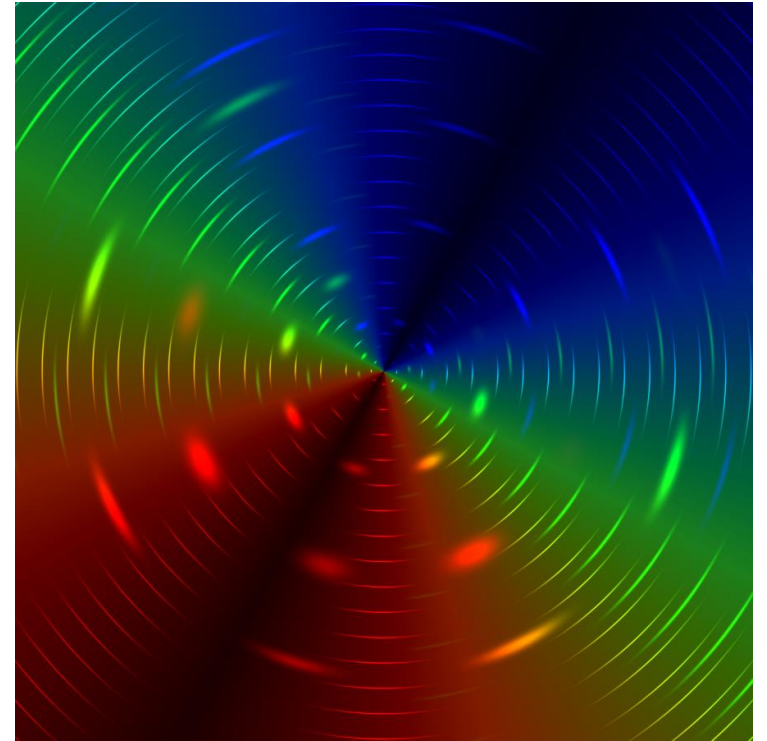
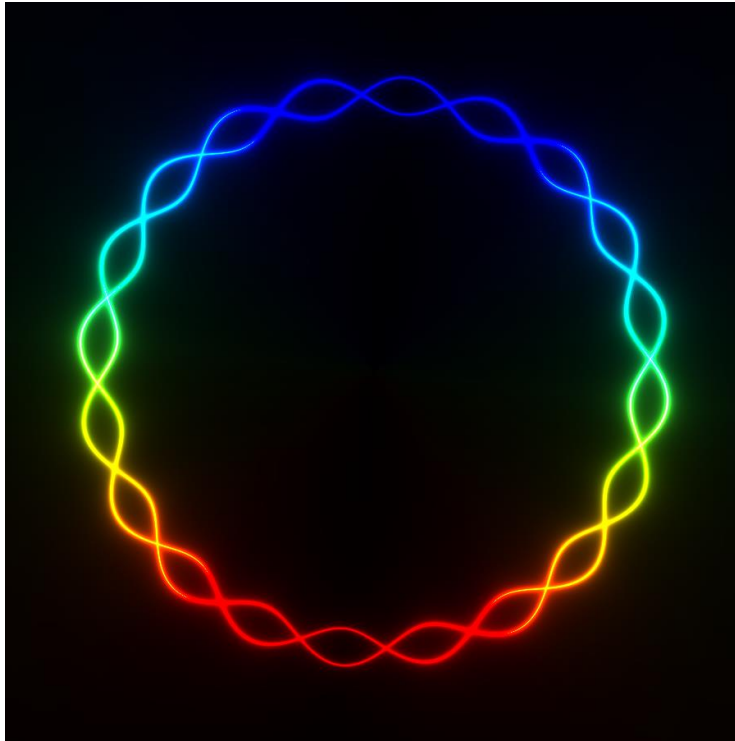
# Pixel Shader

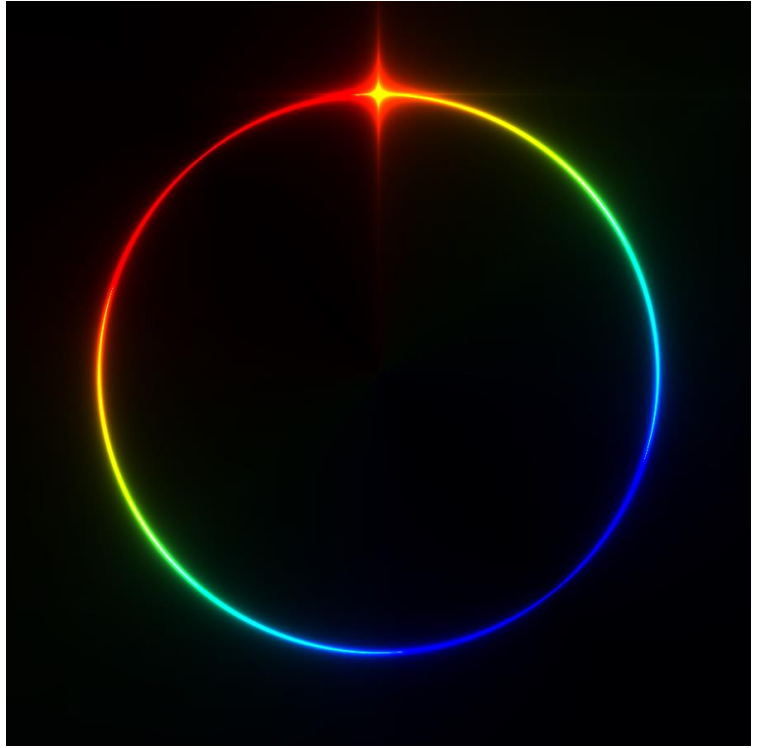
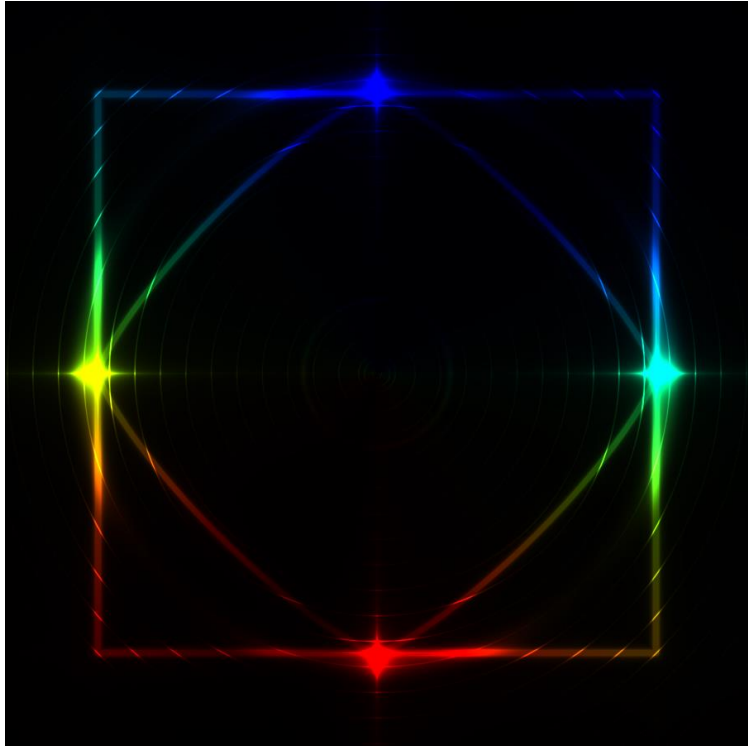
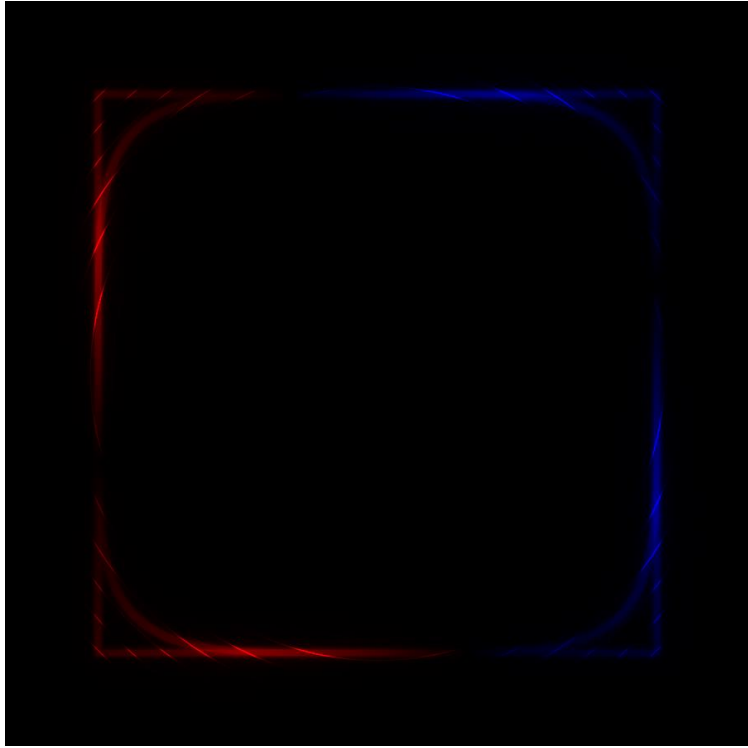
- Przetwarza wszystkie piksele po rasteryzacji
- Zwykle używany do:
  - Obliczania światła padającego na fragment
  - Obliczania koloru fragmentu na podstawie położenia oraz koloru z tekstury
  - Wykonywania innych operacji graficznych (efekty specjalne)
- Jest wykonywany równolegle na wielu fragmentach
- Otrzymuje dane od vertex shader-a





# Przykładowe shadery

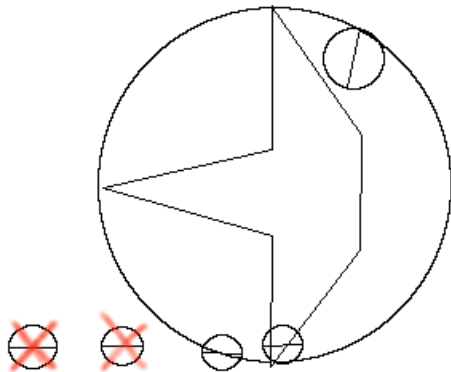




# Wykrywanie kolizji

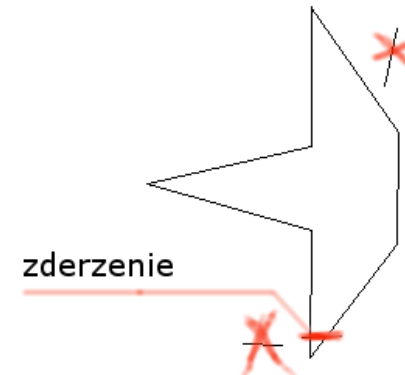
## Broad Phase:

- Wykrywanie możliwych zderzeń między parami obiektów
- Szybkie obliczenia wykluczające większość par



## Narrow Phase:

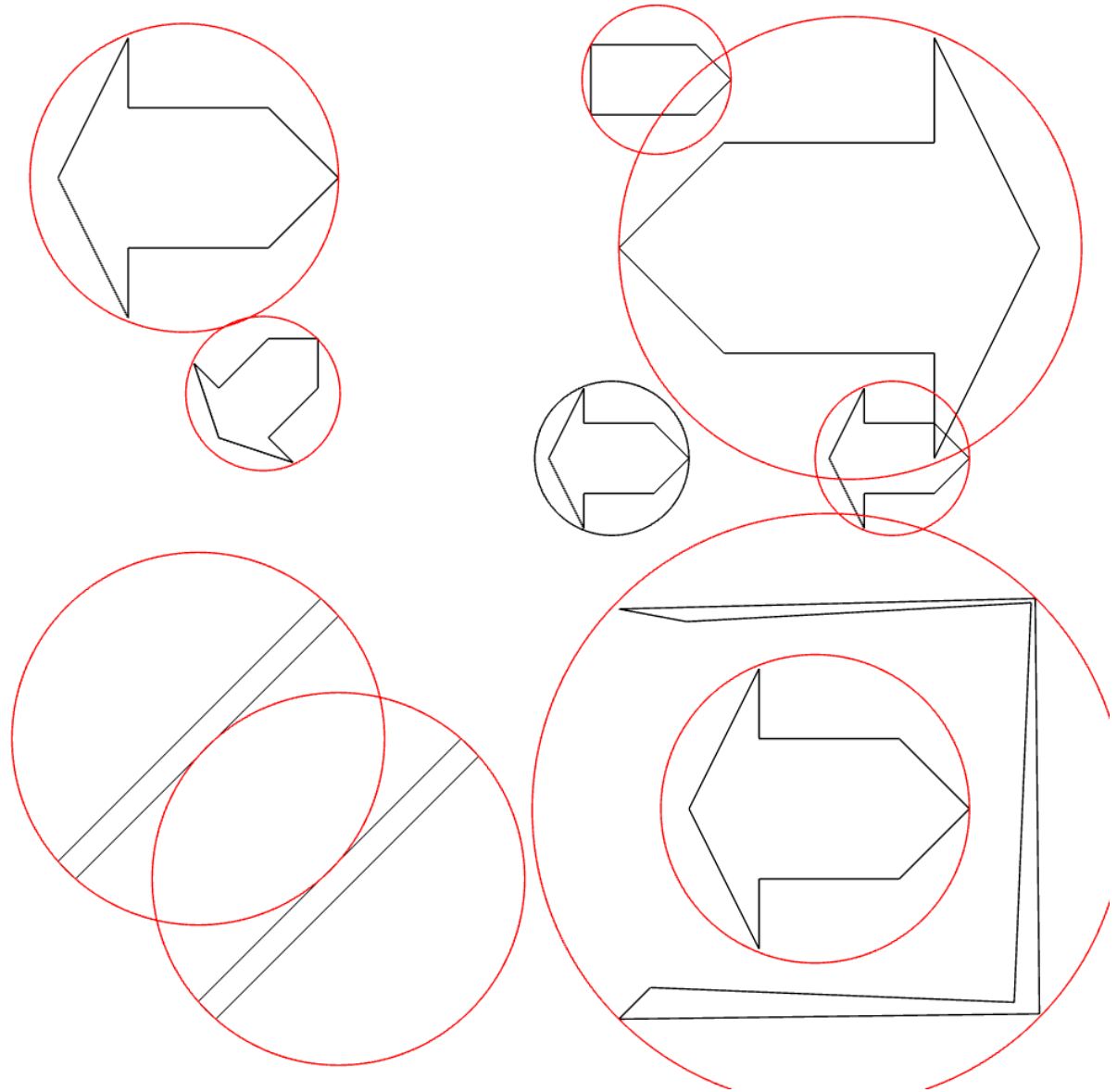
- Dokładne wykrywanie zderzeń
- Dłuższe obliczenia dla par obiektów, które przeszły przez pierwszy etap



# Broad Phase

## Metoda Bounding Circles

- Otoczenie każdego obiektu okręgiem
- Sprawdzenie czy okręgi się pokrywają:
  - Obliczenie odległości pary obiektów
  - Sprawdzenie, czy suma promieni jest mniejsza niż odległość



# Inne metody dla Broad Phase

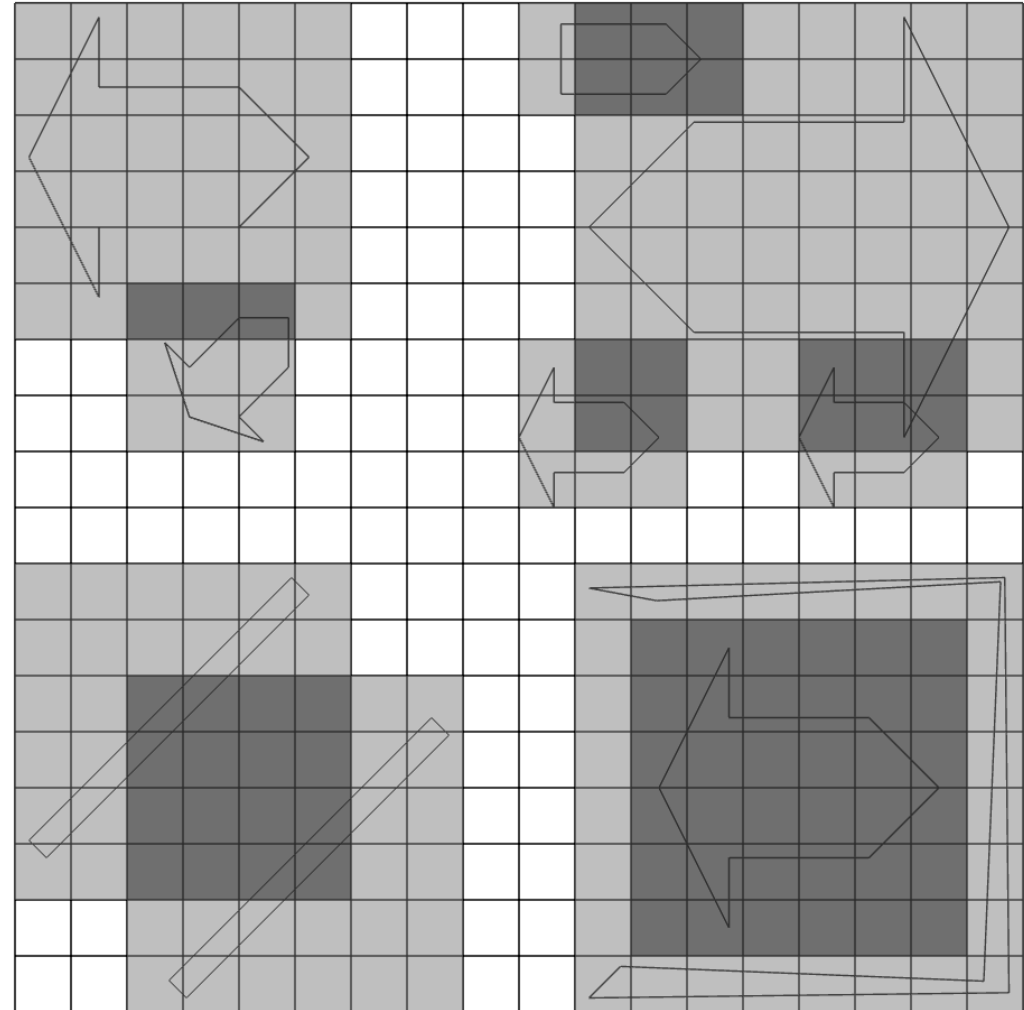
Inne metody Bounding Volumes:

- AABB
- Bounding Rectangle

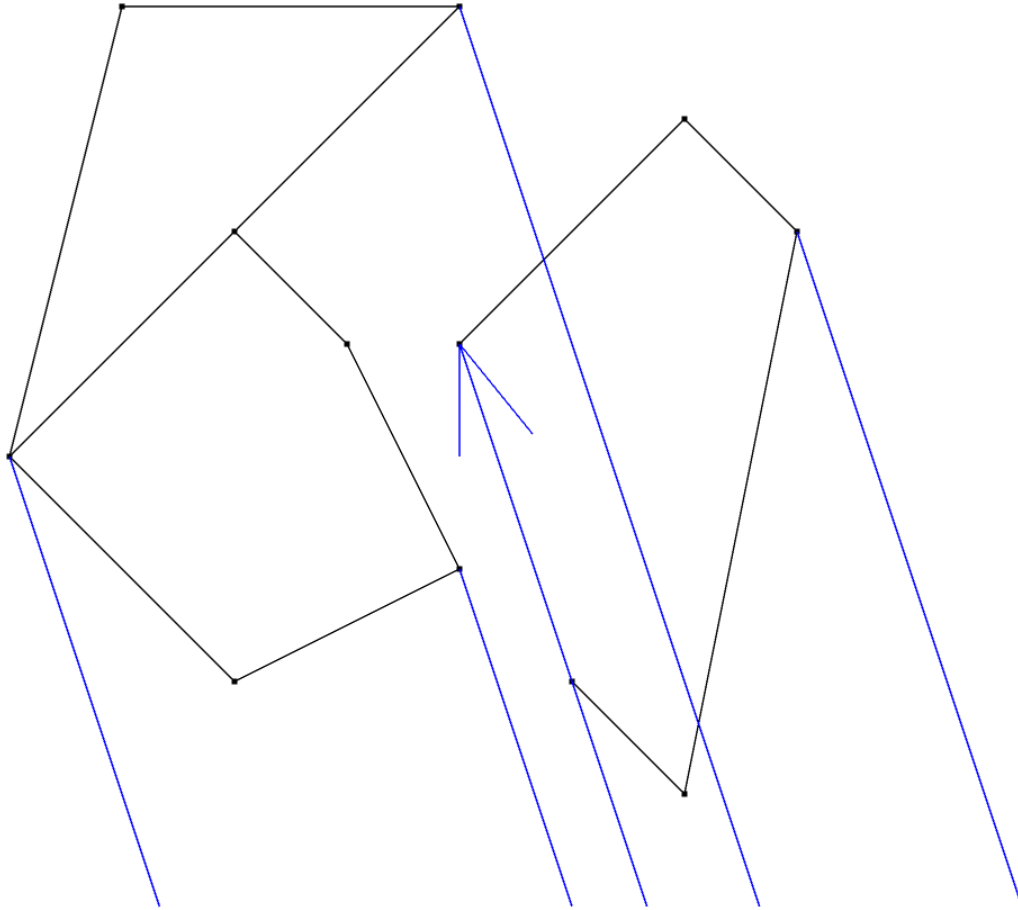
Metoda Sweep and Prune

Metody Space partitioning:

- Bins
- Quadtree/Octree
- Bounding Volume Hierarchy
- Binary Space Partitioning



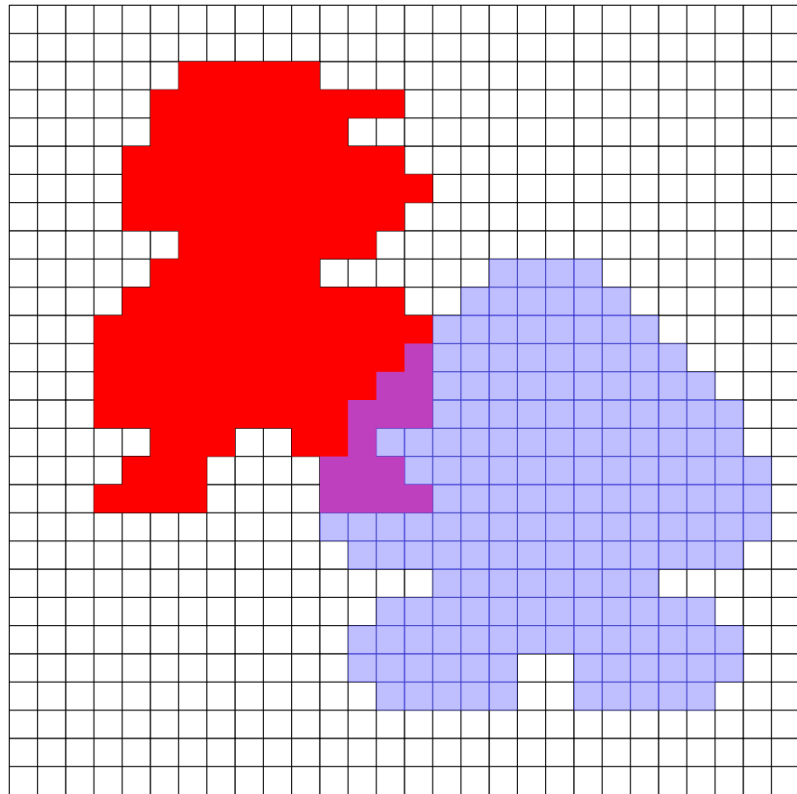
# Narrow phase



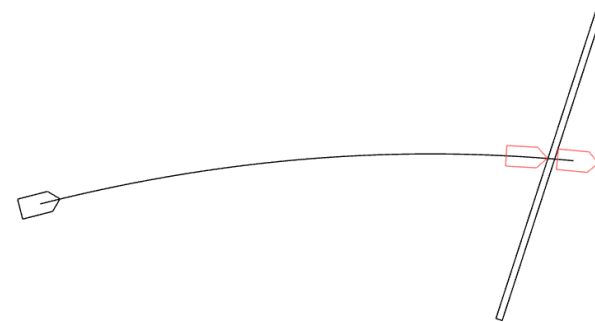
- Metoda Płaszczyzny Oddzielającej
- Dokładnie wykrywa kolizje
- Działa na modelach
- Działa dla modeli będących otoczkami wypukłymi (convex hull)
- Przykład dyskretnego wykrywania kolizji

# Inne metody dla narrow phase

Pixel perfect collision detection



Ciągłe wykrywanie kolizji



# Muzyka

## Część programowa:

- Wczytywanie dźwięków
- Odtwarzanie i łączenie dźwięków
- Obsługa głośności

## Część artystyczna:

- Ścieżka dźwiękowa
- Efekty dźwiękowe



# Inne części gry

- Efekty
- Synchronizacja i kolejność obliczeń:
  - Rysowanie tylko, gdy nie są prowadzone obliczenia
  - Odpowiednia kolejność wywoływania funkcji odświeżających
- Odpowiednie dopasowanie różnych elementów i zbalansowanie gry

Dziękuję za uwagę 😊