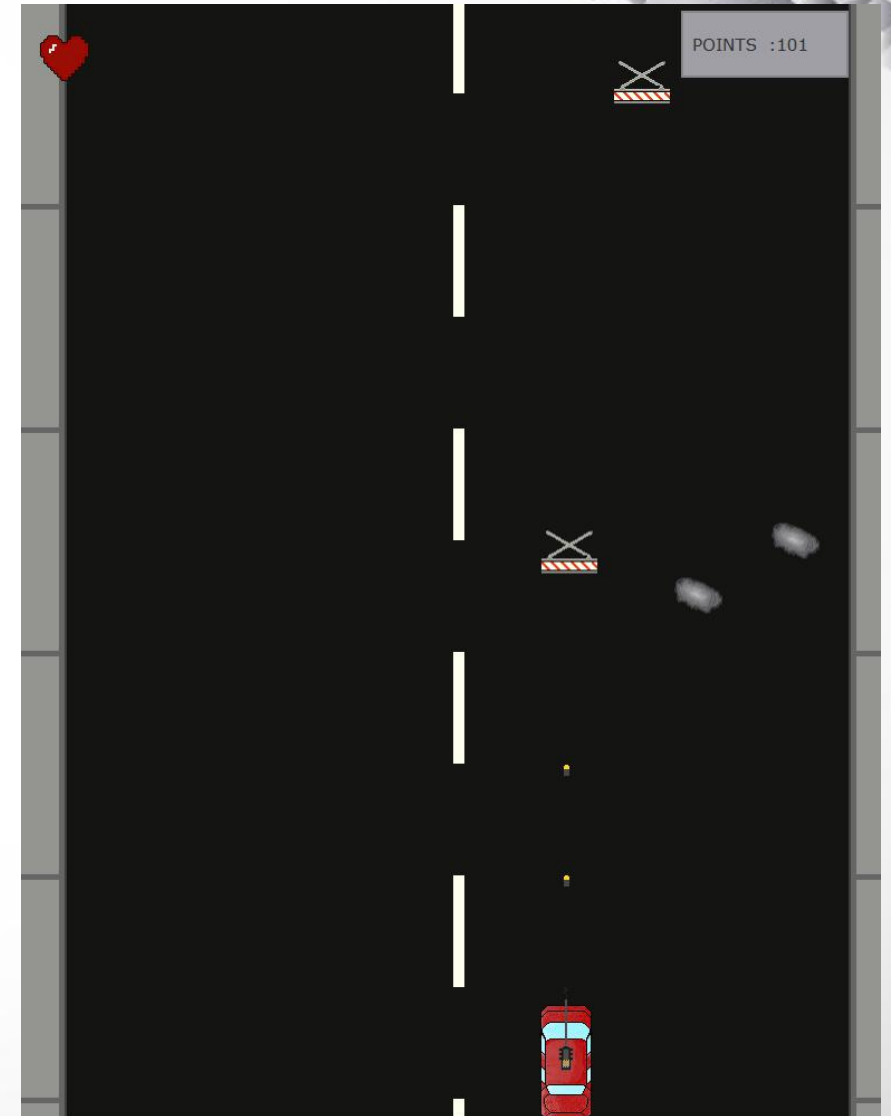


# Inspiracja - gra River Raid z 1982 roku

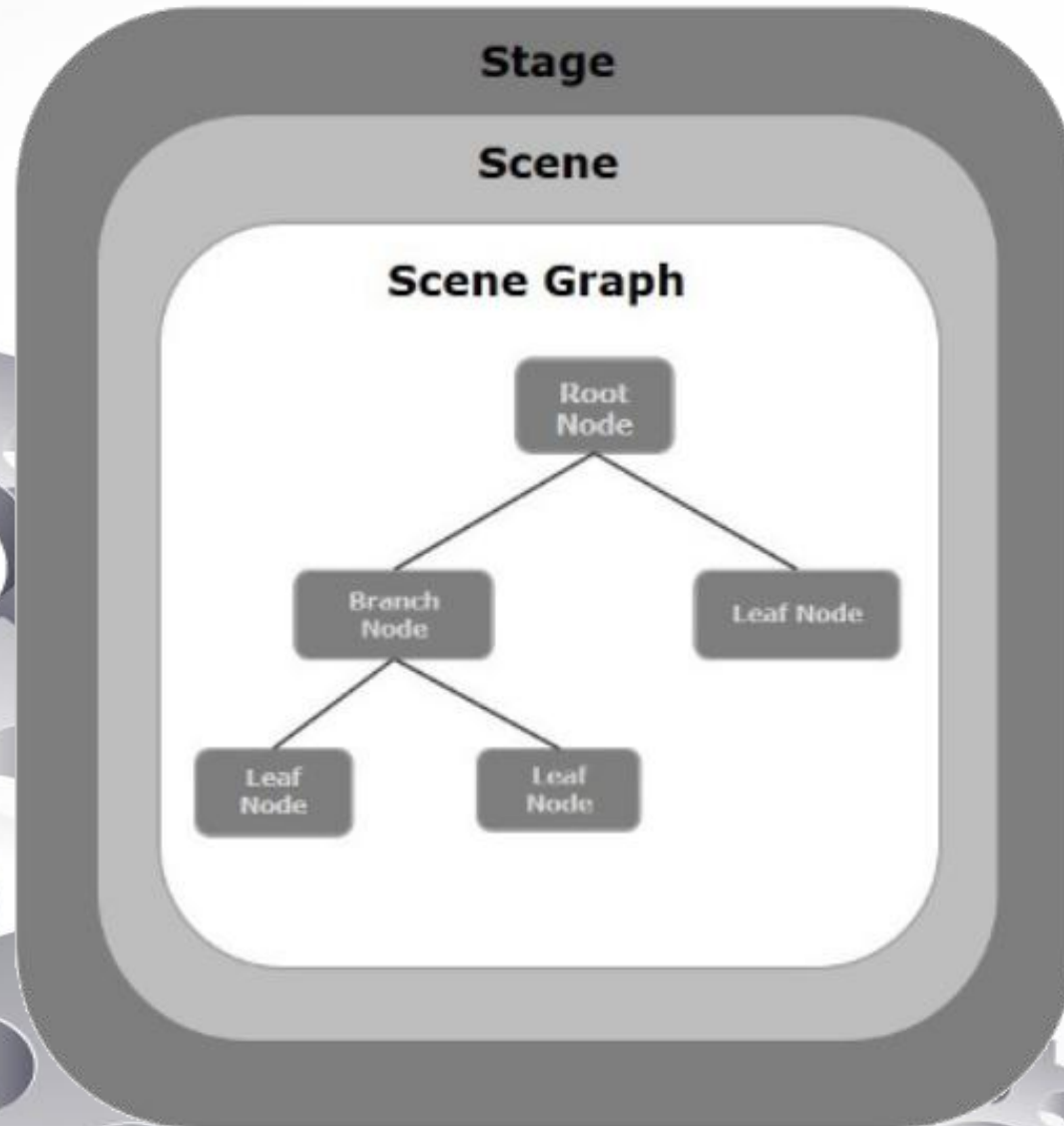


# JavaFX



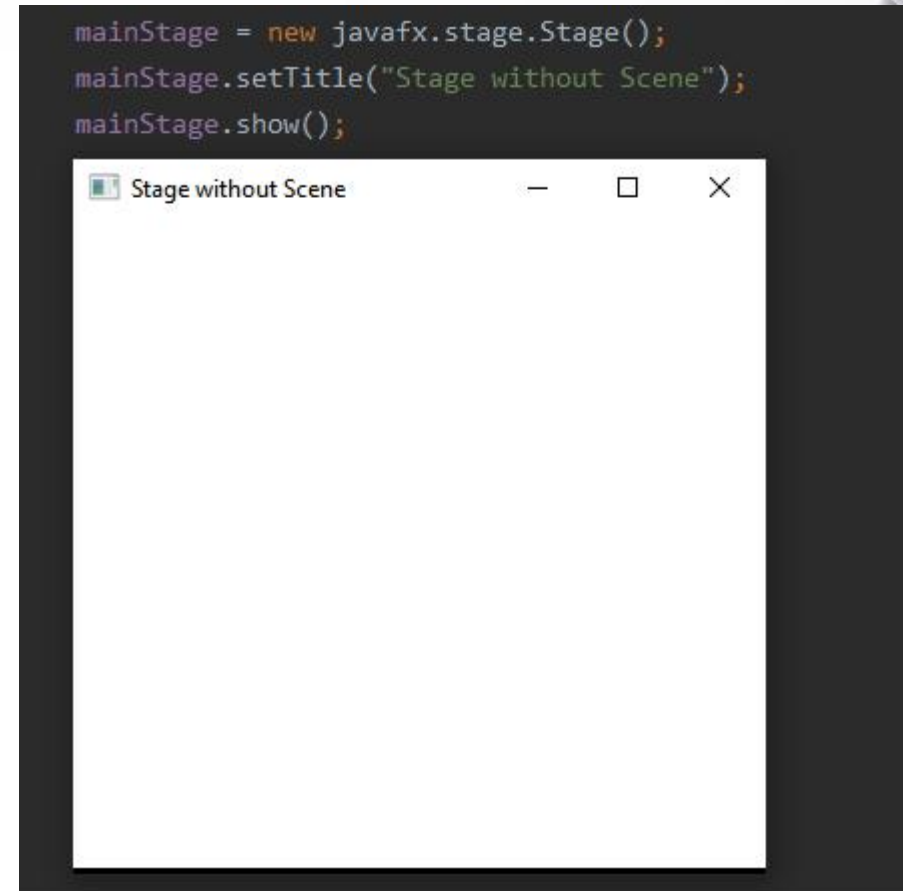
- Technologia oprogramowania, zestaw bibliotek Java
- Główne przeznaczenie to tworzenie tzw. Rich Internet Applications.
- Wieloplatformowość

# Struktura aplikacji JavaFX



# Stage

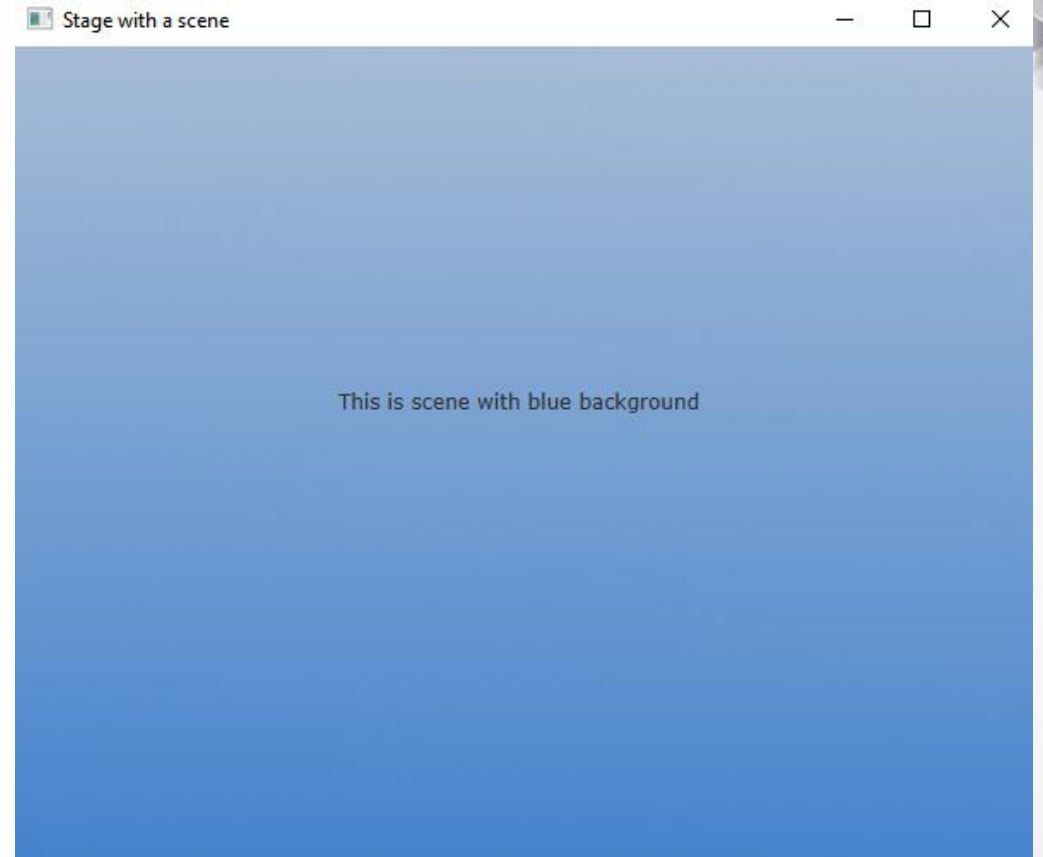
- inaczej window, okno
- reprezentowane przez klasę Stage
- zawiera wszystkie elementy aplikacji JavaFX



Przykładowe okno bez przypisanej sceny

# Scene

- reprezentuje fizyczną zawartość aplikacji
- obiekt klasy Scene może być przypisany tylko do jednego obiektu klasy Stage
- przy tworzeniu sceny do konstruktora należy podać element root - w naszym projekcie są to kontenery AnchorPane lub GridPane



Przykładowa scena z ustawionym tłem i etykietą (klasa Label)

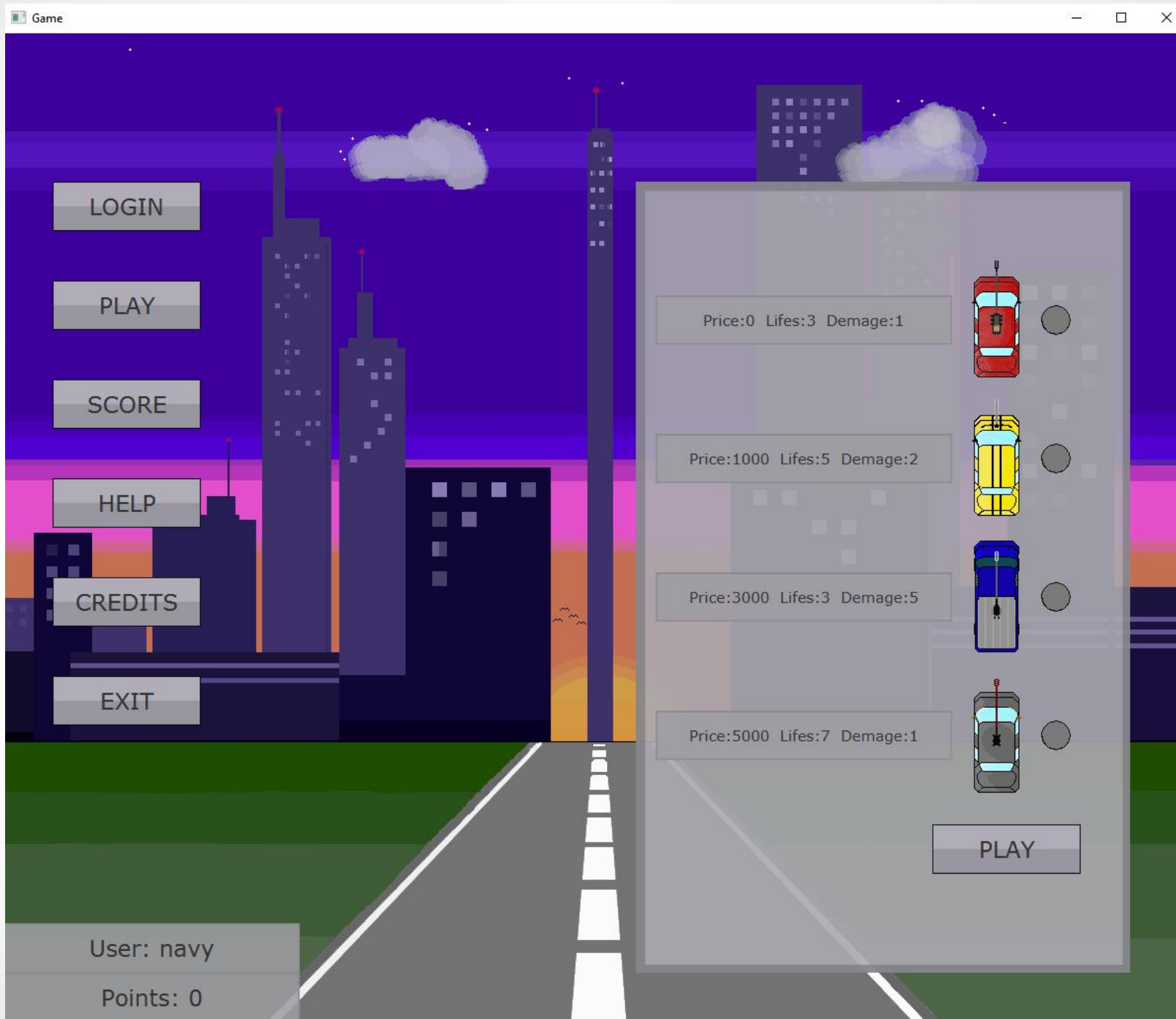
# Nodes



- Może zawierać
  - obiekty geometryczne
  - UI controls - różnego rodzaju przyciski, suwaki, checkboxy, itp
  - **kontenery** - określają rozkład elementów na scenie
  - elementy audio oraz video

```
Label label = new Label( text: "This is scene with blue background");
AnchorPane primaryPane = new AnchorPane();
primaryPane.getChildren().add(label);
label.setLayoutX(190);
label.setLayoutY(200);
label.setFont(Font.font("Verdana"));
```

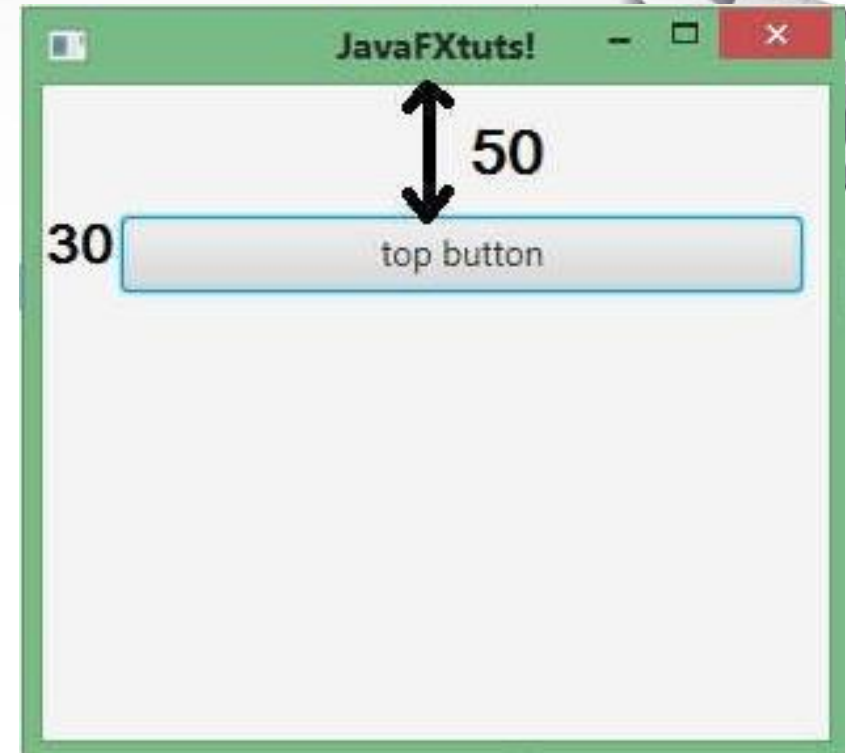
Przykład - dodawanie etykiety do kontenera typu AnchorPane





# Klasa AnchorPane

- jest kontenerem na różnego rodzaju elementy (dzieci) np. przyciski, etykiety itp
- stanowi pewnego rodzaju tło
- pozycja dzieci jest określana na podstawie odległości od krawędzi anchor pana
- umożliwia ustawianie pozycji niewidocznych elementów (metoda `isVisible(false)`)



```
public void createBackground()  
{  
    Image backgroundImage = new Image( url: "view/resources/MAIN_MENU.png", requestedWidth: 1200, requestedHeight: 1000, preserveRatio: false, smooth: true);  
    BackgroundImage background = new BackgroundImage(backgroundImage, BackgroundRepeat.REPEAT, BackgroundRepeat.NO_REPEAT, BackgroundPosition.DEFAULT, size: null);  
    mainPane.setBackground(new Background(background));  
}
```

Przykład - ustawianie tła kontenera a zarazem sceny, do której zostanie przypisany





- Tło

- wykorzystanie klasy GridPane
- Dwa elementy GridPane zawierające obraz tła
- Pierwszy element ustawiony jest w pozycji (0,0)
- Drugi element jest przesunięty o długość obrazka
- Oba elementy poruszają się z określoną prędkością
- **PROBLEM** - rozjeżdżanie się elementów tła, podczas zwiększania tempa gry



- Przeszkody/bonusy

- przeszkody zaimplementowane w postaci tablicy obiektów klasy ImageView
- równoległa tablica typu int - wytrzymałość przeszkód
- przechodzenie po elementach tablic w pętli
- bonusy analogicznie

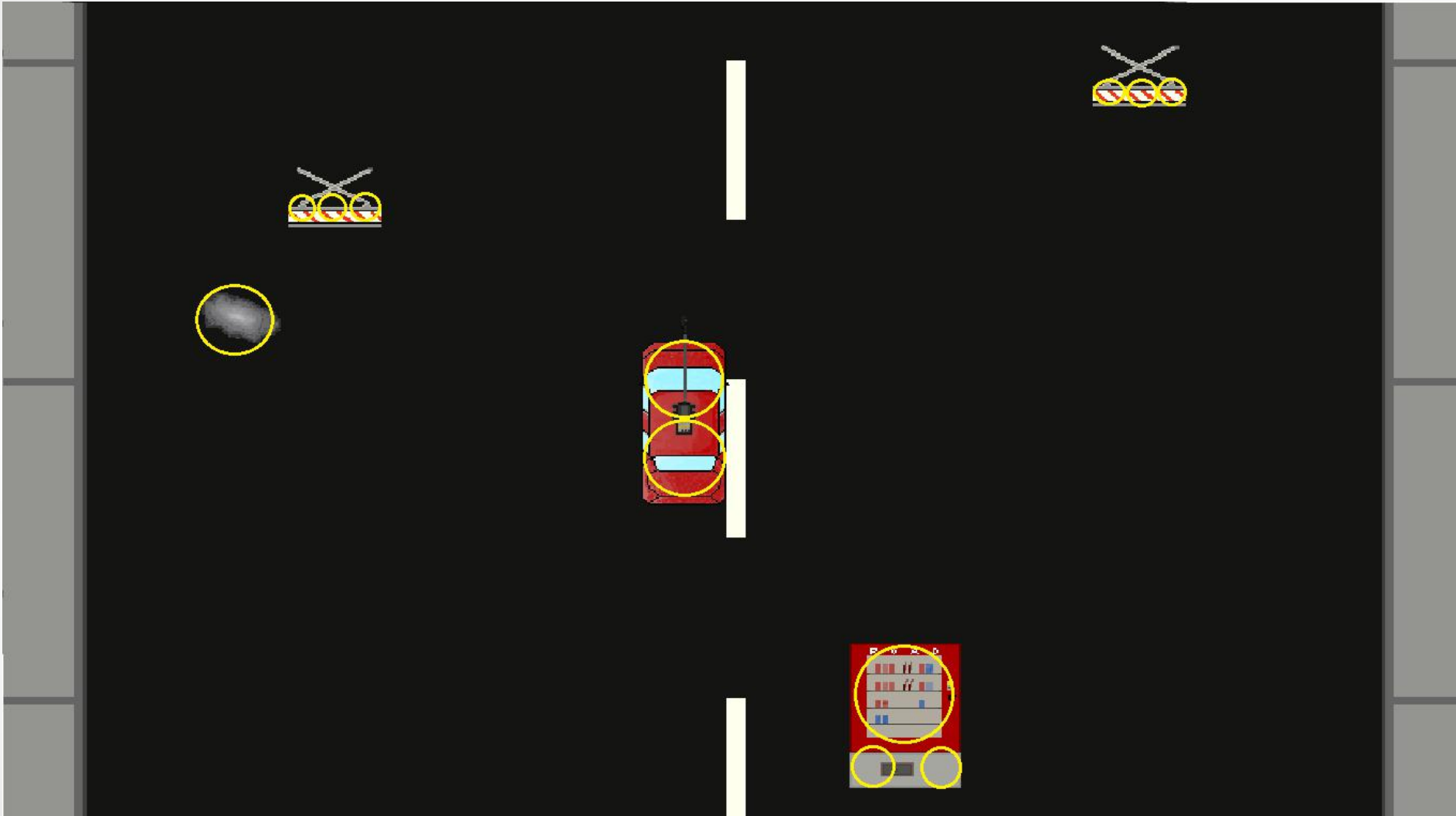


## • Klasa AnimationTimer

- animacje oraz cały ruch gry stworzony za pomocą obiektu klasy AnimationTimer
- metody zawarte w timerze wykonują się w każdej klatce
  - przesuwanie tła
  - sprawdzanie tablic
  - sprawdzanie kolizji samochodu z elementami
  - sprawdzanie kolizji pocisków z elementami
  - przesuwanie wystrzelonych pocisków po ekranie
  - kontrola klawiszy sterowania itp.



# Kolizje - przybliżanie obiektów okręgami



PROBLEM - błąd w kodzie powodował przelatywanie pocisków przez niektóre obiekty = brak wykrywania kolizji  
PROBLEM - konieczne odpowiednie dopasowanie okręgów kolizji względem pozycji obiektu ImageView

# Efekty dźwiękowe - 8bit sound effects



- 8 bit, ponieważ stare konsole były wyposażone w 8 bitowe procesory dźwiękowe
- W naszym projekcie do odtwarzania wykorzystujemy klasę AudioClip
  - odtwarzanie wielu efektów naraz
  - funkcjonalność fire and forget
- **PROBLEM** - niektóre z wybranych przez nas dźwięków były za krótkie i AudioClip ich nie odtwarzał - **POMYSŁ NA ROZWIĄZANIE ??**

# Zapisywanie danych do pliku



- zapisywanie ilości posiadanych punktów (waluta)
- zapisywanie najlepszego wyniku osiągniętego w trakcie pojedynczej rozgrywki
- wykorzystanie hash-mapy przechowującej dane podczas trwania programu
- zapisywanie do pliku w momencie wychodzenia z gry
- przy otwieraniu programu wczytanie danych z pliku

```
private void saveScore()
{
    for (Map.Entry<String, Integer> entry : MainMenu.scoreSave.entrySet()) {
        if (entry.getKey().equals(MainMenu.userName)) {
            if (entry.getValue() < inGamePoints)
                entry.setValue(inGamePoints);
        }
    }
}
```

Fragment kodu - metoda aktualizująca best score

# Koniec, czas na część praktyczną :)



Źródła: <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm>  
[https://www.tutorialspoint.com/javafx/javafx\\_application.htm](https://www.tutorialspoint.com/javafx/javafx_application.htm)