

**Projekt Programistyczny Indywidualny
(2022/2023)**

Space Invaders

Autor:

Michał Szykaruk

1. Wstęp

1.1. Opis projektu

Projekt jest grą typu shoot 'em up, w której gracz kontroluje statek kosmiczny i musi eliminować wrogie statki, unikając jednocześnie kolizji z ich pociskami. Gra została napisana inspirowana Space Invaders. Jest to klasyczna gra arkadowa, która została wydana w 1978 roku przez japońską firmę Taito. Gracz kontroluje statek kosmiczny, który musi bronić Ziemi przed inwazją obcych istot z kosmosu. Celem gry jest zestrzelenie jak największej liczby obcych statków, które stopniowo opadają w kierunku gracza. Gra jest znana ze swojej prostoty, dynamicznej rozgrywki i charakterystycznego dźwięku, którym towarzyszy zestrzeliwanie obcych.

2. Założenia projektowe

2.1. Założenia wstępne podstawowe

- Gra będzie dostępna na komputery osobiste z systemem Windows.
- Gracz będzie mógł kontrolować statek kosmiczny, poruszając się w pionowym i poziomym kierunku strzelając do wrogich statków.
- Wrogie statki będą atakować gracza, próbując go zniszczyć. Gracz będzie musiał unikać kolizji z nimi, jednocześnie eliminując wrogów za pomocą strzałów.
- Projekt będzie oferował dwa tryby rozgrywki: tryb dla jednego gracza, w którym gracz będzie awansował na kolejne poziomy, oraz tryb dla dwóch graczy, w którym dwie osoby będą mogły rywalizować ze sobą.
- Implementacja gry będzie oparta na bibliotece Pygame, która zapewni obsługę grafiki, dźwięku oraz interakcji z użytkownikiem.

2.2. Założenia dodatkowe

- Gra ma posiadać tryb multiplayer.
- Gra ma posiadać menu.
- Gra ma posiadać możliwość sterowania dźwiękiem w zakładce ustawienia.

2.3. Podsumowanie założeń

Założenia zostały w pełni zrealizowane. Każdy z wyżej wymienionych punktów został spełniony.

3. Model programu

- Ekran startowy: Po uruchomieniu gry, gracz zostanie powitany ekranem startowym, gdzie będzie mógł rozpocząć nową grę, dostosować ustawienia lub wyjść z gry.
- Po rozpoczęciu nowej gry, gracz przejmie kontrolę nad statkiem kosmicznym umieszczonym na dolnej części ekranu. Statek będzie poruszał się w pionowym lub poziomym kierunku, a gracz będzie mógł sterować nim za pomocą klawiatury.
- Wrogie statki będą pojawiać się u góry ekranu i będą atakować gracza, strzelając w jego kierunku. Gracz będzie musiał unikać strzałów wrogich statków.
- Gracz będzie miał możliwość strzelania do wrogów, aby je zniszczyć.
- Gra będzie oferować różnorodne grafiki poziomów. Każdy poziom będzie bardziej wymagający od poprzedniego, zapewniając wyzwanie dla gracza.
- Gra będzie prowadziła system punktacji, który będzie śledził osiągnięcia gracza, takie jak aktualny poziom, co jest równoznaczne z ilością punktów.

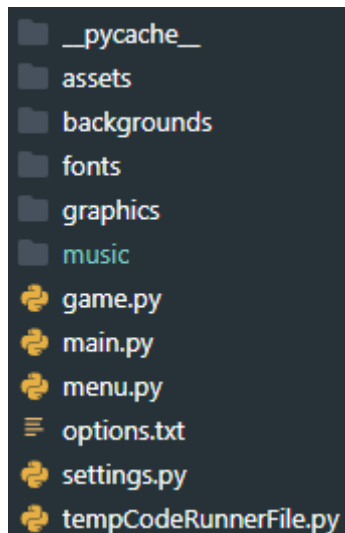
3.1. Przebieg gry

W trakcie rozgrywki gracz kontroluje statek kosmiczny, który znajduje się na dolnej części ekranu. Statek porusza się w pionowym i poziomym kierunku, a gracz może nim sterować przy użyciu klawiatury. Wrogie statki zaczynają pojawiać się z górnej części ekranu i rozpoczynają atak na gracza, strzelając w jego kierunku. Strzały wrogich statków mogą spowodować utratę zdrowia i ewentualnie doprowadzić do zniszczenia statku gracza. Gra kończy się wraz ze śmiercią gracza.

4. Opis

Jest to prosta gra z interfejsem graficznym oparta na bibliotece Pygame. Gra składa się z menu głównego, trybu jednoosobowego i trybu wieloosobowego. Plik `game.py` zawiera logikę gry, a plik `menu.py` odpowiada za wyświetlanie menu i interakcję z użytkownikiem. Plik `main.py` uruchamia menu i jest punktem wejścia do gry. Plik `settings.py` zawiera stałe, które są ważne dla samej rozgrywki.

4.1. Pliki



`main.py`:

- Importuje moduł `pygame`.
- Importuje klasę `Menu` z pliku `menu.py`.
- Sprawdza, czy aktualny moduł jest uruchamiany jako główny moduł.
- Inicjalizuje bibliotekę `pygame`.
- Tworzy instancję klasy `Menu`.
- Wywołuje metodę `main_menu()` na obiekcie `menu`.

game.py:

- Importuje moduły pygame, random i sys.
- Importuje moduł settings.
- Importuje funkcję choice z modułu random.
- Definiuje klasę Game.
- Inicjalizuje obiekt gracza, poziom gry, zmienną przechowującą ilość żyć, grupy obiektów aliens i alien_lasers.
- Zawiera metody do konfiguracji poziomu, wyświetlania statystyk, wykrywania kolizji, tworzenia przeciwników, strzelania przez przeciwników i aktualizowania ekranu.

menu.py:

- Importuje moduł pygame i sys.
- Importuje moduł settings.
- Importuje klasy Game, Player i Player2 z pliku game.py.
- Definiuje klasę Button i klasę Menu.
- Klasa Button reprezentuje przycisk w menu i zawiera metody do wyświetlania przycisku, sprawdzania czy kursor znajduje się nad przyciskiem oraz zmiany koloru przycisku.
- Klasa Menu reprezentuje menu gry i zawiera metody do odtwarzania muzyki, uruchamiania gry w trybie jednoosobowym i wieloosobowym, wyświetlania menu głównego oraz zapisywania i odczytywania opcji gry.

settings.py

- Inicjalizacja biblioteki Pygame i czcionek:
- Początkowe importowanie modułu Pygame oraz inicjalizacja czcionek.
- Ustawienia gry:
- Określenie rozdzielczości gry (1280x720 pikseli).
- Limit klatek na sekundę (60 FPS).
- Obliczenie połowy szerokości i wysokości ekranu.
- Ustawienie koloru czcionki (111, 196, 169).
- Inicjalizacja ekranu gry o określonej rozdzielczości.
- Zegar gry do kontroli czasu.
- Załadowanie tła gry w postaci listy obrazków.
- Muzyka:
- Importowanie plików dźwiękowych używanych w grze.
- Muzyka obejmuje dźwięki związane z przyciskami, tematem głównym, graczem, wrogami i strzałami.
- Ustawienia gracza:
- Określenie prędkości gracza, prędkości pocisków gracza oraz czasu odświeżania gracza.
- Ustalenie początkowej liczby żyć gracza.
- Załadowanie grafik związanych z graczem i pociskami gracza.
- Ustawienia wrogów:
- Określenie prędkości wrogów, prędkości pocisków wrogów oraz czasu odświeżania wrogów.
- Załadowanie grafiki związanej z pociskami wrogów.
- Załadowanie różnych skórek wrogów.

4.2. Opis klas

Klasa Game:

- Odpowiada za główny przebieg gry, inicjalizację, aktualizację i renderowanie obiektów.
- Zarządza stanem gry, takimi jak poziom, wynik i liczba żyć.
- Obsługuje zdarzenia wejściowe, takie jak ruch gracza i strzały.
- Sprawdza kolizje między obiektami i reaguje na nie, np. kolizje pocisków gracza z wrogami.
- Wyświetla interfejs użytkownika, taki jak wynik i liczba żyć.

Klasa Player:

- Reprezentuje gracza w grze.
- Odpowiada za poruszanie się gracza, zarówno w poziomie, jak i w pionie.
- Zarządza strzałami gracza, takimi jak inicjowanie strzałów i ich aktualizacja.
- Monitoruje zdrowie gracza i reaguje na kolizje z wrogimi pociskami.
- Przechowuje grafikę gracza i jego parametry, takie jak prędkość poruszania się.

Klasa Player2:

- Podobnie jak klasa Player, reprezentuje drugiego gracza w grze.
- Odpowiada za poruszanie się drugiego gracza i zarządzanie jego strzałami.
- Monitoruje zdrowie drugiego gracza i reaguje na kolizje.
- Przechowuje grafikę drugiego gracza i jego parametry.

Klasa Laser:

- Reprezentuje pociski w grze, zarówno gracza, jak i wrogów.
- Odpowiada za poruszanie się pocisków i sprawdzanie kolizji z innymi obiektami.
- Przechowuje grafikę pocisków i ich parametry, takie jak prędkość.

Klasa Enemy:

- Reprezentuje wrogów w grze.
- Zarządza ruchem wrogów, takim jak poruszanie się w określonym kierunku.
- Tworzy wrogie pociski i zarządza nimi.
- Monitoruje zdrowie wrogów i reaguje na kolizje.
- Przechowuje grafikę wrogów i ich parametry, takie jak prędkość.

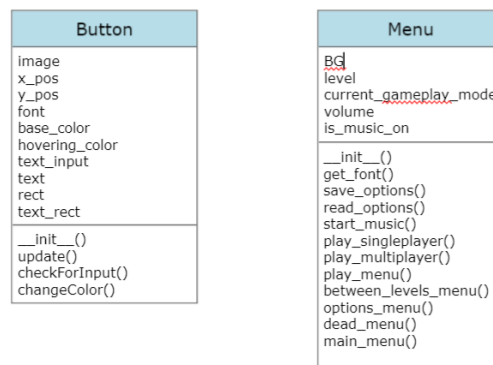
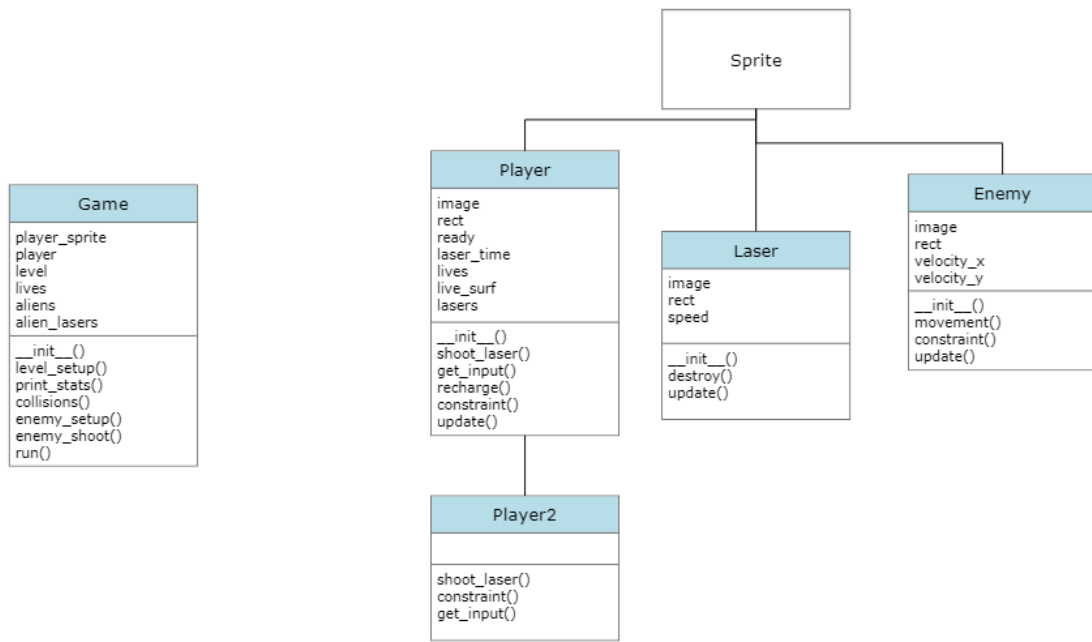
Klasa Button:

- Reprezentuje przyciski w interfejsie użytkownika.
- Odpowiada za ich wygląd, interakcję i reakcje na zdarzenia, takie jak kliknięcie.
- Przechowuje tekst na przycisku, grafikę i pozycję.

Klasa Menu:

- Zarządza menu startowym gry.
- Tworzy i obsługuje przyciski menu.
- Odpowiada za reakcje na wybór opcji, takie jak rozpoczęcie nowej gry lub zakończenie aplikacji.
- Przechowuje grafiki tła menu i inne elementy interfejsu.

4.3 Diagram klas



5. Bibliografia

- Dokumentacja Pygame: [<https://www.pygame.org/docs/>]
- Artykuł o Space Invaders [https://en.wikipedia.org/wiki/Space_Invaders]
- Grafiki statków kosmicznych z [<https://www.flaticon.com>]
- Zdjęcia z Google Images:
[https://www.google.com/search?q=space+1920x1080&client=opera-gx&hs=zgN&sxsrf=APwXEde6J57MvP1aREN9SbltNrlqtjuawQ:1686166006557&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjFkOnu8bH_AhWHuosKHXi_Dq8Q_AUoAXoECAEQAw&biw=1879&bih=939&dpr=1]
- Dostęp do oryginalnego repozytorium dźwięków został utracony w trakcie pisania programu, a pliki były pierwotnie pozyskane z pomocą wyszukiwarki google