



POLITECHNIKA RZESZOWSKA  
im. Ignacego Łukasiewicza  
WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Tomasz Kozioł

Projekt - SPACE LANDER

Rzeszów 2023

# 1. Wstęp

Celem tego projektu jest stworzenie prostej gry 2D, "Space Lander", z użyciem biblioteki Pygame. Gra ta polega na sterowaniu statkiem kosmicznym w taki sposób, aby bezpiecznie wylądował on na powierzchni.

## 2. Opis Problemu

Zadanie polega na sterowaniu statkiem kosmicznym, który jest ciągle pod wpływem siły grawitacji. Gracz może obracać statek i uruchamiać silniki odrzutowe, aby zmienić kierunek i prędkość lotu. Celem gry jest bezpieczne wylądowanie na powierzchni, co jest możliwe tylko przy odpowiednim kącie nachylenia statku, co było najtrudniejszą częścią całego projektu.

### 3. Metodologia

Do stworzenia gry użyliśmy języka Python i biblioteki Pygame, która dostarcza podstawowe narzędzia do tworzenia gier 2D. Zdefiniowaliśmy klasę Ship, która reprezentuje statek kosmiczny, a także funkcje main\_menu() i main\_game(), które odpowiadają za logikę menu głównego i głównej pętli gry.

```
class Ship:
    new *
    def __init__(self, x, y):
        self.position = pygame.math.Vector2(x, y)
        self.velocity = pygame.math.Vector2(0, 0)
        self.ship_img = SHIP_IMG
        self.flame_img = FLAME_IMG
        self.show_flame = False
        self.flame_timer = 0 # Licznik czasu płomienia
        self.flame_duration = 1000 # Czas trwania płomienia w milisekundach
        self.angle = 0
        self.speed = 5
        self.landed = False # Zmienna śledząca, czy statek wylądował
        self.successful_landing = False # Zmienna śledząca, czy lądowanie się powiodło
```

```
def main_game():
    run = True
    FPS = 60

    player = Ship(300, HEIGHT - GROUND_HEIGHT - 50) # Ustalamy początkową pozycję statku na wysokości podłoża

    clock = pygame.time.Clock()

    while run:
        clock.tick(FPS)
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False

        keys = pygame.key.get_pressed()
        if keys[pygame.K_a]: # lewo
            player.rotate_left()
        elif keys[pygame.K_d]: # prawo
            player.rotate_right()

        player.move()

        WIN.blit(BG, (0, 0))
        pygame.draw.rect(WIN, (255, 255, 255), pygame.Rect(0, HEIGHT - GROUND_HEIGHT, WIDTH, GROUND_HEIGHT)) # Rysowanie podłoża
        player.draw(WIN)

        # Wyświetlanie komunikatu o lądowaniu
        if player.landed:
            if player.successful_landing:
                text = "Udane lądowanie!"
            else:
                text = "Nieudane lądowanie. Zbyt duży kąt nachylenia."
            text_surface = FONT.render(text, True, (255, 255, 255))
            text_rect = text_surface.get_rect(center=(WIDTH // 2, HEIGHT // 2))
            WIN.blit(text_surface, text_rect)

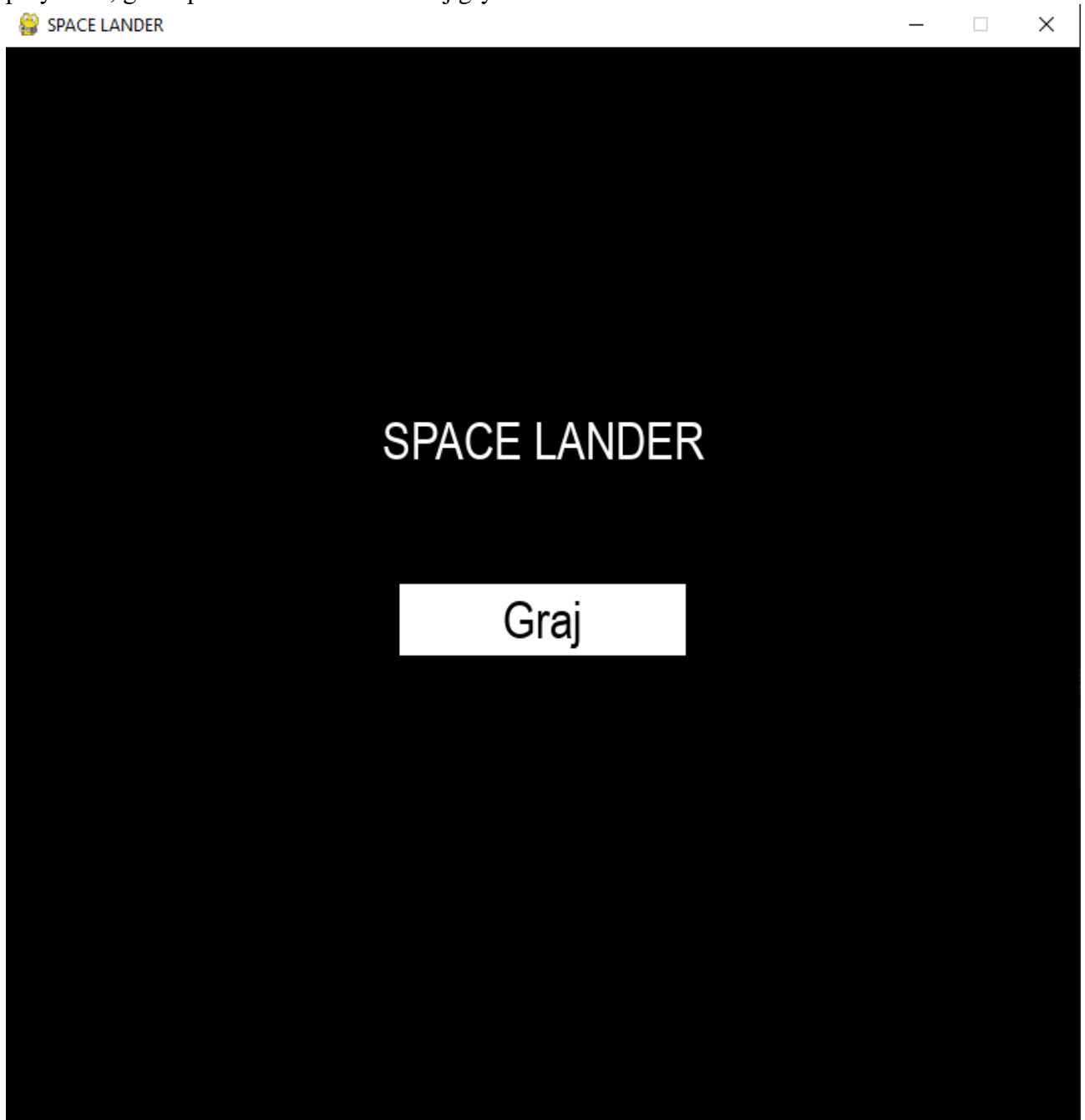
        pygame.display.update()

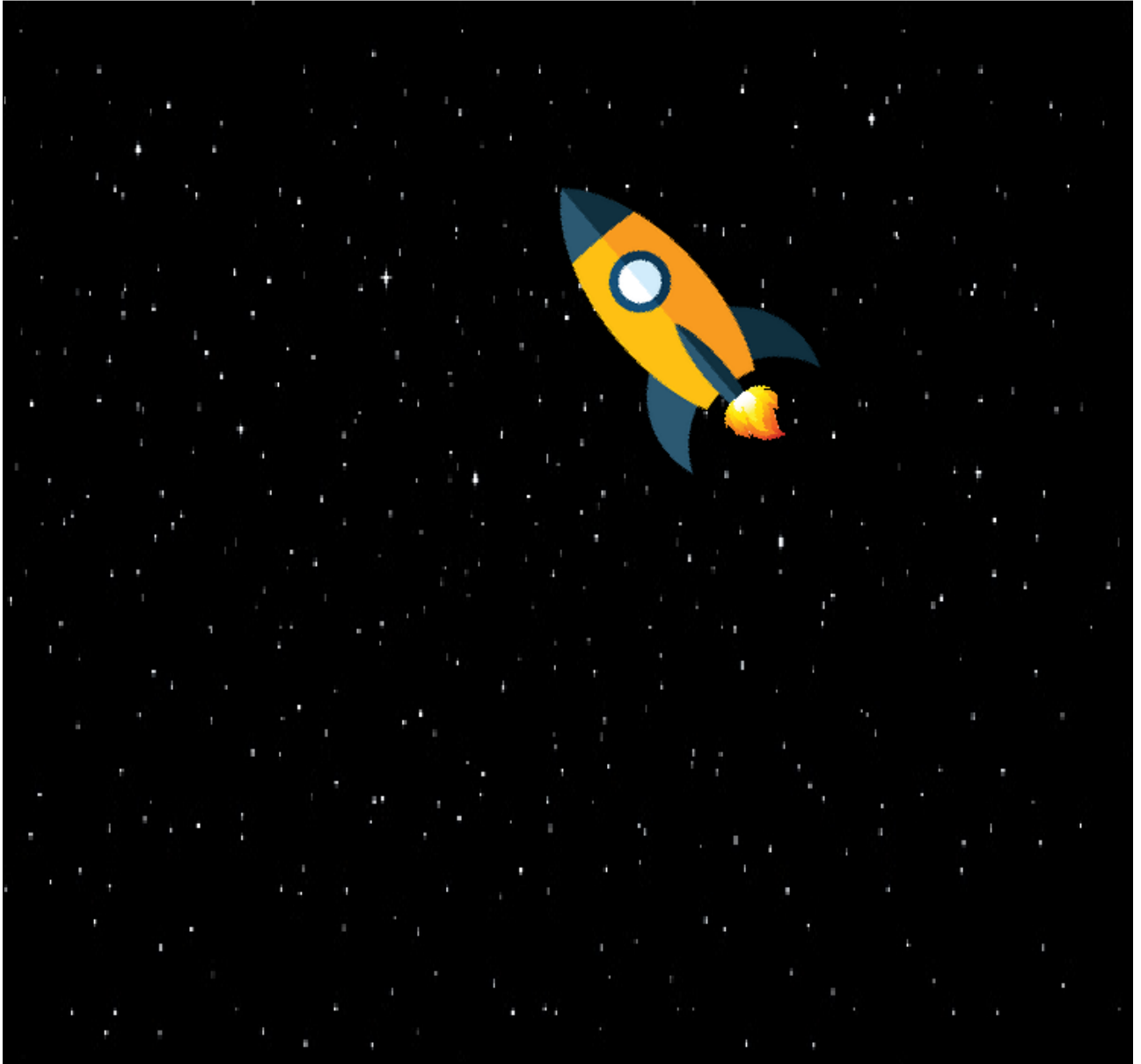
    pygame.quit()
```

## 4. Implementacja

Gra została podzielona na dwie główne części: menu główne i właściwą grę.

Menu główne to prosta pętla, która wyświetla tytuł gry i przycisk "Graj". Po kliknięciu przycisku, gracz przechodzi do właściwej gry.





Właściwa gra to pętla, w której gracz steruje statkiem kosmicznym. Statek może być obracany i może uruchamiać silniki odrzutowe. Jeżeli statek wyląduje na powierzchni z odpowiednim kątem nachylenia i odpowiednią prędkością, lądowanie jest uznawane za udane. W przeciwnym przypadku, lądowanie jest nieudane.

## 5. Podsumowanie

Projekt "Space Lander" pokazuje, jak można stworzyć prostą, ale interesującą grę 2D z użyciem języka Python i biblioteki Pygame. Choć gra jest prosta, wymaga od gracza precyzji i umiejętności pilotażu statku kosmicznego. Projekt ten może być rozwinięty o dodatkowe funkcje, takie jak różne poziomy trudności, różne typy powierzchni do lądowania, czy system punktacji.