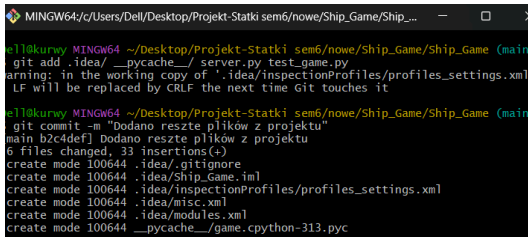


Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
1	UI	JEST	pygame.init()	<input type="checkbox"/>		
		Wprowadzanie danych	input_text += event.unicode (wpisywanie IP)	<input type="checkbox"/>		2
		Wyświetlanie danych	pygame.draw.rect(screen, color, (...)) (siatka, statki)	<input type="checkbox"/>		2
		Zmiana danych	dragging_ship["pos"] = event.pos (przeciąganie statku) dragging_ship.pos = event.pos	<input type="checkbox"/>		2
		Wyszukiwanie danych	get_ship_at_pos(pos) (znajdowanie statku pod kursorem)	<input type="checkbox"/>		2
		Przedstawienie wyników	text = font.render("Wygrałeś!", True, GREEN) (komunikat wygranej)	<input type="checkbox"/>		2
2	Podstawy	Zmienne	WIDTH, HEIGHT = 1400, 800	<input type="checkbox"/>		2
		typy danych	ships = [] (lista statków)	<input type="checkbox"/>		2
		komentarze	# Initialize Pygame # Get local server IP address	<input type="checkbox"/>		1
		operatory	LEFT_BOARD_X = (WIDTH - (GRID_SIZE * CELL_SIZE * 2 + BOARD_GAP)) // 2	<input type="checkbox"/>		1,5
		Instrukcje warunkowe (if, elif, else)	if screen_mode == "connect": if not game_started:	<input type="checkbox"/>		3
		Instrukcje iteracyjne				
		for	for ship in ship_objects: for x in range(0, GRID_SIZE * CELL_SIZE + 1, CELL_SIZE):	<input type="checkbox"/>		2

		while	while running: while len(clients) < 2:	<input type="checkbox"/>		2
		Operacje wejścia (input)	input_text += event.unicode	<input type="checkbox"/>		1,5
		Operacje wyjścia (print)	print(f"Server running on IP: {get_local_ip()}, port: {PORT}")	<input type="checkbox"/>		1,5
		Funkcje z parametrami i wartościami zwracanymi	def place_ship_on_board(ship, cell): def handle_client(client, opponent, player_id):	<input type="checkbox"/>		2
		Funkcje rekurencyjne	count_hits_recursive(shots, index=0)	<input type="checkbox"/>		3
		Funkcje przyjmujące inne funkcje jako argumenty	apply_to_shots(shots, func)	<input type="checkbox"/>		3
		Dekoratory	@log_time	<input type="checkbox"/>		1,5
3	Kontenery	Użycie listy	ships = [] clients = []	<input type="checkbox"/>		2
		Użycie słownika	ship_objects = [{"size": (4, 1), ...}] ship_images = {}	<input type="checkbox"/>		2
		Użycie zbioru	forbidden_zone = set()	<input type="checkbox"/>		1,5
		Użycie krotki	ship["size"] = (4, 1)	<input type="checkbox"/>		1,5
4	Przestrzenie nazw	Zastosowano zmienne lokalne	x, y = ship["pos"]	<input type="checkbox"/>		1,5
		Zastosowano zmienne globalne	global placing_ships, game_started	<input type="checkbox"/>		1,5
		Zastosowano zakresy funkcji	data = pickle.loads(client.recv(1024))	<input type="checkbox"/>		1,5
		Zastosowano zakresy klas	self.size w klasie Ship	<input type="checkbox"/>		1,5
5	Moduły i pakiety	Projekt podzielony na moduły (import, __init__)	import pygame, socket, threading, pickle, os from game import reset_game_state	<input type="checkbox"/>		2

Nr	Obszar	Wymaganie	KOD		Przyznane pkt	Pkt max
		Własne pakiety/funkcje pomocnicze w osobnych plikach .py	server.py, game.py, test_game.py	<input type="checkbox"/>		2
6	Obsługa błędów	Obsługa wyjątków (try, except, finally)	try: client_socket.connect((ip, PORT)) except Exception as e: try: ship_images = {...} except FileNotFoundError as e:	<input type="checkbox"/>		2
		Użycie assert do testów i walidacji	self.assertTrue(result)	<input type="checkbox"/>		1,5
7	Łańcuchy znaków	Operacje na stringach (m.in. formatowanie, dzielenie, wyszukiwanie)	print(f"Server running on IP: {get_local_ip()}, port: {PORT}")	<input type="checkbox"/>		2
8	Obsługa plików	Odczyt z plików .txt, .csv, .json, .xml (min. 1)	pygame.image.load(os.path.join(IMAGE_FOLDER, "ship_4_h.png"))	<input type="checkbox"/>		2
		Zapis do plików .txt, .csv, .json, .xml (min. 1)	plt.savefig("game_stats.png")	<input type="checkbox"/>		2
9	OOP	Klasy	class Ship: class BattleShip(Ship): class Game:	<input type="checkbox"/>		2
		Metody	def rotate(self): def place(self, cell): def use_ability(self):	<input type="checkbox"/>		2
		Konstruktory	def init(self, size, pos, orientation="h"):	<input type="checkbox"/>		2
		Dziedziczenie	class BattleShip(Ship):	<input type="checkbox"/>		2
10	Programowanie funkcyjne	map	list(filter(lambda shot: shot[1], shots))	<input type="checkbox"/>		1,5
		filter	list(filter(lambda shot: shot[1], shots))	<input type="checkbox"/>		1,5
		lambda	lambda x, y: (x[0] + y[0], x[1] + y[1])	<input type="checkbox"/>		1,5
		reduce	reduce(add_coords, [shot[0] for shot in shots], (0, 0))	<input type="checkbox"/>		1,5

11	Wizualizacja danych	Wygenerowano wykres (np. matplotlib, seaborn)	plt.bar(["Moje strzały", "Strzały przeciwnika"], hits, color=['blue', 'red'])	<input type="checkbox"/>		2
		Zapisano wykres do pliku graficznego (.png lub .jpg)	plt.savefig("game_stats.png")	<input type="checkbox"/>		1,5
12	Testowanie	Testy jednostkowe (assert, unittest, pytest)	class TestGameLogic(unittest.TestCase):	<input type="checkbox"/>		1,5
		Testy funkcjonalne		<input type="checkbox"/>		1,5
		Testy Integracyjne		<input type="checkbox"/>		1,5
		Testy graniczne / błędne dane	def test_place_out_of_bounds_ship(self):	<input type="checkbox"/>		1,5
		Testy wydajności (np. czas wykonania, timeit)	duration = timeit(lambda: Ship((1, 1), (0, 0)).place((0, 0)), number=1000)	<input type="checkbox"/>		1,5
		Testy pamięci memory_profile		<input type="checkbox"/>		1,5
		Test jakości kodu (flake8, pylint)		<input type="checkbox"/>		1,5
13	Wersjonowanie	Repozytorium GIT	git init	<input type="checkbox"/>		1
		Historia commitów	git log	<input type="checkbox"/>		1
Nr	Obszar	Wymaganie	KOD 		Przyznane pkt	Pkt max
		Link do GitHub	https://github.com/Maxym0909	<input type="checkbox"/>		1
		Opis commitów	git commit -m "Dodano resztę plików w projekcie"	<input type="checkbox"/>		1
14	Dokumentacja	Plik README.md (cel, autorzy, uruchamianie)	(Zawarte w PDF i Google Docs: cel, autorzy, uruchomienie)	<input type="checkbox"/>		1,5

Dr inż. Dariusz Michalski. Formularz samooceny do projektu z języków skryptowych

		Przykładowe dane wejściowe i wyjściowe	text_lines = ["Zasady gry w Statki:", ...]	<input type="checkbox"/>		2
		Diagram klas lub struktura modułów		<input type="checkbox"/>		2
		SUMA				

Dokumentacja znajduje się w poniższym linku wraz z opisem przebiegu całej gry:

<https://docs.google.com/document/d/1KWYJIDvjchTBXqGG5wnprJfAsJGtM9hpPBxMnyxGsog/edit?tab=t.0>