

Raport z Testów Systemowych i Jednostkowych

Projekt: GardenPlanner (Symulator Ogrodu)

1. Wstęp

Celem niniejszego dokumentu jest przedstawienie wyników testów oprogramowania GardenPlanner. Testy miały na celu weryfikację poprawności działania kluczowych funkcjonalności, stabilności systemu oraz spójności danych. Proces obejmował testy manualne (funkcjonalne/UI) oraz testy automatyczne (logika biznesowa).

2. Środowisko Testowe

- **System operacyjny:** Windows 11
- **Silnik:** Godot Engine 4.3 (.NET Edition)
- **Baza danych:** MySQL 8.0 (Docker Container)

3. Strategia i Dobór Danych

Do testów dobrano dane reprezentatywne, obejmujące przypadki skrajne (brzegowe) oraz typowe.

Dane wejściowe:

- *Rośliny:* Róża (mała, wymagająca słońca), Dąb (duży, rzucający cień), Igłak (zimozielony).
- *Gleby:* Piaszczysta (słaba retencja), Gliniasta (wysoka retencja).
- *Warunki:* Zima (miesiąc 12), Lato (miesiąc 6).

4. Wyniki Testów Automatycznych (Unit Tests)

Testy automatyczne zostały zaimplementowane w skrypcie `TestRunner.cs` i weryfikowały logikę matematyczną symulacji.

ID	Nazwa Testu	Opis Przypadku	Oczekiwany Wynik	Wynik
AT-01	TestGrowthCalculation	Obliczenie wzrostu dla idealnych warunków (dobra gleba + woda).	Współczynnik > 1.0	✓ PASS
AT-02	TestBadSoilConditions	Obliczenie wzrostu dla złych warunków (niedobór wody).	Współczynnik < 1.0	✓ PASS
AT-03	TestWinterGrowthStop	Sprawdzenie modyfikatora sezonowego dla miesiąca 12 (Zima).	Modyfikator == 0.0	✓ PASS
AT-04	TestShadowLogic	Obliczenie nasłonecznienia, gdy obok stoi wyższa roślina.	SunLevel < 1.0	✓ PASS
AT-05	TestDatabaseFallback	Symulacja braku połączenia z DB.	Użycie InitialData	✓ PASS

Tabela 1: Wyniki testów automatycznych

5. Wyniki Testów Manualnych (Systemowych)

5.1 Funkcjonalności UI i Interakcja

ID	Scenariusz	Dane Testowe	Rezultat Oczekiwany	Rezultat Rzeczywisty	Status
MT-01	Sadzenie rośliny (Drag&Drop)	Wybór „Róza”, kliknięcie w polu ogrodu.	Roślina pojawia się w miejscu kursora.	Zgodny.	✓ PASS
MT-02	Sadzenie poza granicami	Próba posadzenia poza zdefiniowanym obszarem 10x10m.	Zablokowanie akcji, komunikat w konsoli.	Zgodny.	✓ PASS
MT-03	Zmiana prędkości czasu	Kliknięcie przycisku „Prędkość” (1x -> 2x).	Timer przyspiesza, licznik miesięcy rośnie szybciej.	Zgodny.	✓ PASS
MT-04	Ruch kamery (Pan/Zoom)	Scroll myszy, PPM + przesunięcie.	Widok przybliża się/oddala i przesuw.	Zgodny.	✓ PASS

Tabela 2: Testy funkcjonalności UI

5.2 Integracja i Dane

ID	Scenariusz	Opis	Status
IT-01	Zapis stanu (JSON)	Zapisanie ogrodu z 5 roślinami w 15. miesiącu. Plik garden_save.json powstaje.	✓ PASS
IT-02	Odczyt stanu (JSON)	Wczytanie pliku po restarcie aplikacji. Rośliny wracają na pozycje z zachowanym wiekiem.	✓ PASS
IT-03	Ładowanie z MySQL	Uruchomienie z aktywnym Dockerem. Dane pobrane z tabeli Roslina.	✓ PASS

Tabela 3: Testy integracji

6. Statystyka Znalezionych Błędów

W trakcie cyklu testowego wykryto łącznie 5 defektów.

Priorytet	Opis Błędu	Status Naprawy	Komentarz
Krytyczny	Aplikacja crashuje przy braku pliku .theme.	Naprawiony	Zastąpiono ładowanie pliku generowaniem motywu w kodzie C#.
Wysoki	Rośliny rosną w nieskończoność po przekroczeniu MaxHeight.	Naprawiony	Dodano Mathf.Min w logice wzrostu.
Średni	Brak tabeli ZasadyWzrostu w bazie danych powodował błędy logiki.	Naprawiony	Kod dostosowano do logiki opartej na TypeId w C# (hardcoded rules).
Średni	Iglaki nie rosły w zimie (brak logiki Evergreen).	Odroczony	Funkcjonalność przesunięta do wersji 2.0 (decyzja biznesowa).
Niski	Cień rzucany przez roślinę na samą siebie.	Naprawiony	Dodano warunek if (plantA == plantB) continue;.

Tabela 4: Rejestr błędów

Podsumowanie Statystyczne:

Znaleziono: 5
Naprawiono: 4
Odroczono: 1
Skuteczność napraw: 80%

7. Wnioski Końcowe

Oprogramowanie GardenPlanner przeszło pomyślnie testy akceptacyjne. Kluczowe funkcjonalności (symulacja, zapis, UI) działają stabilnie. System zabezpieczony jest przed błędami krytycznymi (np. brak bazy danych). Zaleca się wdrożenie do produkcji (prezentacji końcowej).