Dokumentacja

# 1. Opis problemu

Na rynku brakuje prostych gier rolniczych w stylu low-poly, które są przystępne dla gracza casualowego i jednocześnie oferują progresję opartą o ekonomię gry. Moja gra ma wypełnić tę lukę, dostarczając lekkiej, przyjemnej rozgrywki bez zbytniej komplikacji i ogromnych map.

# 2. Cele aplikacji

• Stworzenie gry rolniczej o grafice low-poly.

• Umożliwienie graczowi rozwoju gospodarstwa poprzez zakup maszyn.

• Stworzenie prostego systemu ekonomicznego opartego na sprzedaży plonów.

• Zapewnienie przyjemnej i relaksującej rozgrywki.

# 3. Kontekst aplikacji (użytkownicy, powiązane systemy)

**Użytkownicy:**

Gracze casualowi, miłośnicy gier farmerskich, dzieci i młodzież. Gra ma być przystępna i intuicyjna, z prostą mechaniką rozwoju gospodarstwa.

**Powiązane systemy:**

Gra będzie tworzona w silniku Unity z wykorzystaniem Visual Studio do pisania skryptów w języku C#. Modele 3D będą przygotowywane w Blenderze. Wszystkie dane gracza (np. postęp, stan konta, posiadane maszyny) będą zapisywane lokalnie na urządzeniu użytkownika.

# 4. Funkcjonalność

• Start z podstawowymi maszynami: mały ciągnik, pług, siewnik, kombajn.

• Mała mapa z jednym skupem plonów.

• Ekonomia gry: sprzedaż plonów za walutę w grze, którą wydajemy na nowe maszyny.

• Prosty system uprawy: orka → siew → zbiór → sprzedaż.

# 5. Wymagania jakościowe i inne

• Grafika w stylu low poly.

• Intuicyjny interfejs.

• Niski próg wejścia.

• Stabilność działania.

# 6. Wizja konstrukcyjna

Gra będzie zbudowana w oparciu o silnik Unity przy użyciu języka C# oraz z wykorzystaniem modeli 3D przygotowywanych w Blenderze. Główne komponenty gry zostaną podzielone na modularne elementy, aby ułatwić rozbudowę i utrzymanie projektu.

**Silnik gry:**

• Unity – do zarządzania logiką gry, fizyką, animacjami, UI i systemem scen.

• Visual Studio – do pisania skryptów w języku C#.

**Assety i grafika:**

• Wszystkie modele 3D (maszyny, budynki, rośliny) tworzone w Blenderze w stylu low-poly.

• Proste tekstury (stylizowane, pastelowe lub jednolite kolory).

• UI– czytelny, z dużymi ikonami i panelami.

**Maszyny jako prefabrykaty:**

• Każda maszyna (np. traktor, kombajn, pług) będzie oddzielnym prefabem Unity zawierającym:

o Model 3D,

o Collider,

o Skrypt kontroli (np. jazda, uruchamianie maszyny),

o Punkty zaczepienia dla dodatkowego sprzętu (np. przyczepy, pług),

o Animacje (np. obrót kół, ruch narzędzi roboczych).

**System upraw i plonów:**

• Prosty cykl upraw: orka → siew → wzrost → zbiór → sprzedaż.

• Każdy etap obsługiwany przez inną maszynę i prefab.

• Wewnętrznie każdy segment pola będzie posiadał przypisany stan reprezentowany za pomocą typu enum, co ułatwi kontrolowanie logiki upraw oraz zmianę wyglądu terenu w zależności od etapu.

**System ekwipunku i waluty:**

• Gracz będzie posiadał:

o Stan konta (waluta w grze).

o Ilość zebranych plonów.

• Wszystko przechowywane lokalnie.

• Interfejs sklepu umożliwi kupowanie i sprzedawanie maszyn.

**Świat gry:**

• Mała, zamknięta mapa zawierająca:

o Kilka pól uprawnych,

o Jeden budynek skupu plonów,

o Startowe gospodarstwo.

**System zapisu:**

• Dane gracza będą zapisywane lokalnie.

• Przechowywany będzie m.in. stan konta, maszyny, zawartość ekwipunku oraz etap cyklu upraw.

# 7. Ograniczenia i strategia realizacji projektu

**Ograniczenia projektu:**

• Mapa gry zostanie ograniczona do jednej, niedużej lokacji zawierającej kilka pól uprawnych, podstawowe budynki oraz teren umożliwiający swobodną nawigację maszyn rolniczych.

• Projekt realizowany jest samodzielnie, co oznacza brak zespołu specjalistów (grafików, programistów, testerów), a więc konieczność dostosowania skali gry do możliwości jednej osoby.

• Niewielka ilość dostępnych maszyn możliwych do kupienia.

**Strategia realizacji:**

Projekt będzie rozwijany zgodnie z podejściem MVP (Minimum Viable Product), czyli:

**Faza 1 – MVP** (wersja grywalna):

• Mała mapa z kilkoma polami,

• Podstawowe maszyny (ciągnik, pług, siewnik, kombajn),

• Prosty system upraw (orka → siew → zbiór → sprzedaż),

• Sklep ze sprzedażą plonów i zakupem nowych maszyn,

• Prosty interfejs użytkownika (UI),

• Lokalny zapis danych.

**Faza 2 – Testy MVP:**

• Samodzielne testowanie i obserwacja błędów,

• Weryfikacja płynności rozgrywki i intuicyjności interfejsu.

**Faza 3 – Rozszerzenia** (jeśli czas pozwoli):

• Nowe maszyny, typy upraw, ulepszony sklep,

• Powiększenie mapy,

• Ulepszony system ekonomiczny i rozwój gospodarstwa,

• Integracja z platformami (opcjonalnie, np. wersja mobilna).

# 8. Zagrożenia projektu

• Problemy z optymalizacją w Unity.

• Trudność w zaprojektowaniu dobrego balansu ekonomii gry.

# 9. Kryteria osiągnięcia sukcesu

• Gra działa bez błędów krytycznych.

• Umożliwia podstawową rozgrywkę (pług → siew → zbiór → sprzedaż).

• Gracz może kupować nowe maszyny.

• Użytkownicy testowi są zadowoleni z mechaniki gry i stylu graficznego.

# 10. Harmonogram prac nad grą FarmPro

01.04.2025 – 30.04.2025

🔹 Praca nad assetami 3D

• Wyszukiwanie i tworzenie własnych modeli 3D (maszyny, budynki, elementy otoczenia)

• Przygotowanie grafik zgodnych ze stylem low poly

01.05.2025 – 15.05.2025

🔹 Tworzenie mapy gry

• Zaprojektowanie i zbudowanie mapy w Unity

• Rozmieszczenie pól uprawnych, gospodarstwa, budynku skupu i innych elementów

16.05.2025 – 30.06.2025

🔹 Implementacja podstawowych mechanik rozgrywki

• Stworzenie mechaniki poruszania się maszyn po mapie

• Dodanie funkcji orania pola (zmiana tekstury po przejechaniu pługiem)

• Implementacja systemu siewu i wzrostu roślin

• Mechanika zbioru plonów (kombajn zbiera i zapisuje ilość zebranych zasobów)

01.07.2025 – 30.09.2025

🔹 Ekonomia i interakcje gracza

• Wprowadzenie waluty i systemu sprzedaży plonów

• Implementacja budynku skupu, gdzie gracz może sprzedać zebrane plony

• Dodanie interfejsu użytkownika (UI) z portfelem gracza

• Stworzenie sklepu z maszynami (zakup lepszych maszyn)

• System dnia i nocy

• Wprowadzenie systemu zapisu stanu gry

01.10.2025 – 31.10.2025

🔹 Optymalizacja i ulepszenia

• Naprawa błędów wykrytych podczas testów

• Optymalizacja modeli, tekstur i wydajności gry

• Dodanie dźwięków maszyn oraz otoczenia

• Dodanie prostych animacji (np. obrót kół, ruch narzędzi)

01.11.2025 – 01.01.2026

🔹 Testowanie i finalne poprawki

• Pełne testy rozgrywki

• Weryfikacja działania wszystkich systemów

• Usuwanie błędów i dopracowanie detali przed ukończeniem projektu

# 11. Słownik pojęć

**MVP (Minimum Viable Product) –** najprostsza możliwa wersja produktu, która działa i spełnia podstawowe założenia, ale nie zawiera wszystkich funkcji i dodatków**.**

**• Prefab –** gotowy, zapisany obiekt w silniku gry (np. Unity), który można wielokrotnie wstawiać do gry. Może zawierać model, animacje, skrypty i inne elementy**.**

**• Skup –** miejsce w grze, gdzie gracz może sprzedać zebrane plony.

**• Plony –** rośliny (np. zboże), które można uprawiać i sprzedawać w grze.

**• UI (User Interface) –** interfejs użytkownika, czyli wszystko, co gracz widzi na ekranie i z czym może wchodzić w interakcję (np. przyciski, paski, menu), ale nie jest bezpośrednio częścią świata gry.

**• Assety –** wszystkie elementy, z których zbudowana jest gra, takie jak grafiki, dźwięki, modele 3D, animacje, teksty czy czcionki. Twórca gry używa ich do tworzenia świata gry i interfejsu.

**• Collider –** niewidzialna granica wokół obiektu w grze, która służy do wykrywania zderzeń i kontaktów z innymi obiektami.