

# Symulator robota koszącego

*Dokumentacja wstępna*  
*30 października 2025*

## Opis projektu

Projekt zakłada stworzenie w C++ symulatora robota koszącego. Aplikacja umożliwi użytkownikowi zdefiniowanie obszaru do skoszenia, a kosiarka zwizualizuje proces koszenia w czasie rzeczywistym.

## Plan implementacji

Aby zapewnić wydajne działanie aplikacji i uniknąć 100% obciążenia procesora, kluczowym założeniem jest rozdzielenie logiki symulatora od jej wizualizacji. Logika symulacji, odpowiedzialna za generowanie trasy, będzie działać w dedykowanym wątku z pętlą o stałym kroku czasowym (**fixed timestep**), precyzyjnie kontrolowaną przez **std::this\_thread::sleep\_until**. Niezależny wątek GUI, zrealizowany przy użyciu biblioteki graficznej **Qt** lub **SFML**, będzie asynchronicznie odpytywał o stan symulacji. Do zarządzania wątkami i podstawowej synchronizacji posłużą **standardowa biblioteka C++**, natomiast biblioteka **Boost** zostanie włączona do projektu, aby w razie potrzeby umożliwić implementację innych mechanizmów zarządzania wątkami. System logowania działań kosiarki do pliku tekstowego zaimplementujemy korzystając z **std::ofstream**. Pliki kompilowane będą za pomocą **CMake**.

## Funkcjonalności

- Implementacja prostokątnego trawnika, jako ciągłej przestrzeni.
- Użytkownik może skorzystać z metod kosiarki i własnych wskaźników, aby samodzielnie określić obszar do wykoszenia
- Podstawowe funkcje robota: włączanie/wyłączanie mechanizmu koszenia, obrót o dowolny kąt z dokładnością do 1 stopnia, przejazd o zadaną odległość.
- Wykrywanie próby opuszczenia wyznaczonego obszaru i sygnalizowanie tego zdarzenia do programu sterującego.
- Interfejs graficzny (GUI) przedstawiający w czasie rzeczywistym proces koszenia.
- Możliwość przyspieszenia upływu czasu symulacji.
- Zapisywanie szczegółowych logów z akcji podejmowanych przez robota do pliku tekstowego.

Opcjonalnie:

- Wprowadzenie do symulatora efektów dźwiękowych, np. odgłosów pracy kosiarki.
- Gromadzenie i prezentowanie statystyk dotyczących procesu koszenia.

# Lista zadań do realizacji

Poniżej załączamy listę przewidzianych zadań oraz oszacowany czas potrzebny na ich wykonanie.

Zadanie	Szacowany czas wykonania (w godzinach)
Utworzenie struktury katalogów w projekcie	2
Automatyzacja kompilacji za pomocą CMake	3
Plik konfiguracyjny zawierający stałe programowe	2
Stworzenie klas bazowych (np. <i>Lawn</i> , <i>LawnField</i> , <i>Mower</i> )	5
Implementacja mechaniki ruchu i obrotu kosiarki	5
Orientacja kosiarki względem położenia znaczników	5
Detekcja próby wyjścia za granice środowiska	3
Powstrzymanie wyjścia poza granice środowiska	3
Implementacja procesu koszenia	6
Stworzenie GUI	5
Implementacja wyboru kształtu i wyznaczania znaczników	5
Szczegółowe zaprojektowanie wielowątkowości	6
Wprowadzenie usypiania wątku symulacji	5
Wprowadzenie komunikacji pomiędzy wątkami	6
Zaprojektowanie oprawy graficznej symulacji	6
Wizualizacja stanu trawnika	6
Wizualizacja położenia kosiarki	5
Wizualizacja znaczników użytkownika	4
Przyspieszenie czasu wykonania symulacji	3
Implementacja systemu logowania działań kosiarki	2
<i>Opcjonalnie:</i> Implementacja systemu zbierającego statystyki koszenia	3
<i>Opcjonalnie:</i> Prezentacja statystyk w GUI	5
<i>Opcjonalnie:</i> Oprawa dźwiękowa	6
<b>Suma godzin:</b>	<b>101</b>