

UNIwersYTET RZESZOWSKI
Kolegium Nauk Przyrodniczych



Piotr Pasierb

123005

Informatyka i Ekonometria

**Wykorzystanie technologii wirtualnej rzeczywistości do stworzenia
aplikacji szkoleniowej umożliwiającej naukę budowy silnika samochodowego**

Praca inżynierska

Praca wykonana pod kierunkiem
dr Michała Kępskiego

Rzeszów, 2025

Spis treści

1.	Wstęp	5
1.1.	Wprowadzenie	5
1.2.	Cel	6
1.3.	Problem badawczy	7
1.4.	Zakres pracy	7
2.	Podstawowe informacje o budowie silnika samochodowego	9
2.1.	Kluczowe komponenty silnika	9
2.2.	Cykl czterosuwowy	10
3.	Metodyka pracy oraz użyte urządzenia i oprogramowanie	12
3.1.	Metodyka pracy	12
3.2.	Wykorzystane urządzenia	12
3.3.	Wykorzystane oprogramowanie	13
3.4.	Zbieranie i przetwarzanie danych pomiarowych	13
4.	Opis świata rzeczywistego	15
4.1.	Opis zasobów ludzkich	15
4.2.	Dane techniczne	15
4.3.	Wymagania funkcjonalne i нефункционалне	15
5.	Diagramy UML	17
5.1.	Diagram przypadków użycia	17
5.2.	Diagram aktywności	18
5.3.	Diagram klas	20
5.4.	Diagram sekwencji	21
5.5.	Diagram stanów	22
6.	Interfejs użytkownika	24
7.	Analiza i omówienie wyników pracy	28
7.1.	Metodologia analizy	28
7.2.	Statystyki opisowe	28
7.3.	Wpływ wiedzy motoryzacyjnej	29
7.4.	Efekt doświadczenia VR	30
7.5.	Analiza skupień użytkowników	31
8.	Wnioski końcowe	33
9.	Udoskonalenia aplikacji	35
10.	Bibliografia	36
11.	Spis rysunków	37

12. Spis tabel.....	38
13. Streszczenie pracy	39
14. Abstract	41

1. Wstęp

1.1. Wprowadzenie

Współczesny przemysł motoryzacyjny wymaga precyzyjnych i kosztowo efektywnych metod szkoleniowych. Tradycyjne szkolenia mechaników, oparte na fizycznych prototypach silników, wiążą się z wysokimi kosztami materiałów oraz ograniczonym dostępem do specjalistycznych komponentów. Niniejsza praca powstała w odpowiedzi na tę potrzebę, łącząc technologię wirtualnej rzeczywistości (VR) z innowacyjnym podejściem do nauki montażu silników. Głównym celem aplikacji jest nie tylko symulacja realistycznych warunków montażu, ale także dostarczenie narzędzi do obiektywnej oceny kompetencji użytkowników poprzez pomiar czasu oraz interaktywne instrukcje krok po kroku.

Aplikacja oparta na silniku Unity i technologii XR Interaction Toolkit [3], umożliwia użytkownikom składanie silnika w przestrzeni 3D w oparciu o modele utworzone z wykorzystaniem autorskich grafik opracowanych w programie Blender przy zachowaniu rzeczywistych komponentów. Kluczowe funkcje obejmują: inteligentne gniazda montażowe, które weryfikują poprawność połączeń części, dynamiczne instrukcje wyświetlane w formie tekstowej oraz system pomiaru czasu na poziomie pojedynczych części i grup.

W następujących rozdziałach, na które podzielona została praca znajdują się dokładniejsze omówienia poszczególnych elementów pracy.

Rozdział drugi przybliży konstrukcję silników spalinowych w oparciu o dokumentację serwisową silnika Mitsubishi serii 6G7.

W rozdziale trzecim prezentowana jest metodyka utworzenia projektu z uwzględnieniem używanych urządzeń oraz narzędzi.

Rozdział czwarty opisuje rzeczywiste zastosowanie utworzonej aplikacji, w tym wymagania sprzętowe oraz funkcjonalne i нефunkcjonalne.

W zawartości rozdziału piątego znajdują się diagramy UML przedstawiające strukturę aplikacji oraz znajdujące się w niej rozwiązania, które umożliwią dokładniejsze zrozumienie funkcjonowania zrealizowanego systemu.

Rozdział szósty omawia interfejs graficzny użytkownika.

W rozdziale siódmym znajduje się analiza wyników osiągniętych przez użytkowników w trakcie testowania aplikacji oraz zależności z nimi związanymi.

Rozdział ósmy zawiera wyciągnięte wnioski z uzyskanych rezultatów.

Rozdział dziewiąty prezentuje propozycje rozwojowe aplikacji, w tym rozszerzenia funkcjonalności przewidziane do implementacji w kolejnych wersjach systemu.

Przedstawione rozwiązanie prezentuje potencjalne narzędzie dla przemysłu motoryzacyjnego umożliwiające redukcję kosztów, zwiększenie bezpieczeństwa oraz standaryzację procesu edukacji.

1.2. Cel

Głównym celem pracy było stworzenie aplikacji VR wspierającej naukę montażu silnika samochodowego poprzez odwzorowanie kluczowych etapów tego procesu w środowisku wirtualnym. Aplikacja jest oparta na technologii Unity i XR Interaction Toolkit, pozwala użytkownikom na interaktywne składanie silnika w oparciu o modele 3D, z zachowaniem fizyki realistycznych komponentów. Jej nadrzędnym zadaniem jest zapewnienie bezpiecznego i kontrolowanego środowiska szkoleniowego, które eliminuje ryzyko błędów kosztownych w rzeczywistych warunkach.

Centralnym elementem systemu jest mechanizm sekwencyjnego odblokowywania grup części, zarządzany przez klasę `EngineAssemblyManager`. Użytkownik rozpoczyna od całego procesu od bloku silnika, a po jego poprawnej instalacji otrzymuje dostęp do kolejnych elementów tj. tłoki, wał rozrządu itp. Każda część może zostać zamontowana wyłącznie w przeznaczonym dla niej gnieździe, co weryfikuje system walidacji oparty na warstwach interakcji oraz identyfikatorach grup. Próba połączenia niewłaściwych elementów zakończy się niepowodzeniem ze względu na brak możliwości umieszczenia części w gnieździe, które nie jest przeznaczone dla jej grupy.

Aplikacja rejestruje szczegółowe dane o postępach użytkownika, w tym czas montażu poszczególnych części oraz całych sekcji silnika. Wyniki są sortowane i wyświetlane w formie bloków tekstu na przeznaczonym do tego ekranie.

Dodatkowo, dynamiczne instrukcje tekstowe, aktualizowane w czasie rzeczywistym, prowadzą użytkownika krok po kroku, minimalizując ryzyko utknięcia na trudniejszych etapach.

1.3. Problem badawczy

Głównym problemem badawczym pracy jest utworzenie aplikacji VR do nauki montażu silnika, która skutecznie wspomogłoby tradycyjne metody szkoleniowe oparte na fizycznych prototypach. Kluczowe wyzwania obejmują:

1. Utworzenie modeli 3D części silnika.
2. Stworzenie mechanizmu weryfikacji poprawności montażu w czasie rzeczywistym, eliminującego ryzyko błędów w rzeczywistym warsztacie.
3. Zapewnienie płynnej interakcji z częściami silnika w środowisku VR, uwzględniającej fizykę chwytania i przesuwania obiektów.
4. Opracowanie systemu pomiaru czasu i precyzji, który dostarczy możliwość oceny kompetencji użytkowników.

1.4. Zakres pracy

Realizacja pracy obejmowała następujące etapy:

1. Badanie istniejących rozwiązań VR w szkoleniach technicznych oraz zasad budowy silników samochodowych.
2. Projektowanie i tworzenie modeli 3D za pomocą programu Blender w oparciu o dokumentację serwisową Mitsubishi 6G7.
3. Implementacji mechaniki montażu w silniku Unity z wykorzystaniem XR Interaction Toolkit.
4. Opracowanie systemu weryfikacji poprawności montażu z wykorzystaniem identyfikatorów grup i gniazd.
5. Stworzenie modułu pomiaru czasu i sortowaniem wyników według numeracji grup i części.
6. Projekt dynamicznego interfejsu użytkownika z instrukcjami tekstowymi i panelem statystyk czasu.

7. Przeprowadzenie testów funkcjonalnych z udziałem użytkowników, weryfikujących płynność działania i intuicyjność interakcji razem z zebraniem danych do analizy.
8. Przeprowadzenie analizy zebranych danych i utworzenie raportu.

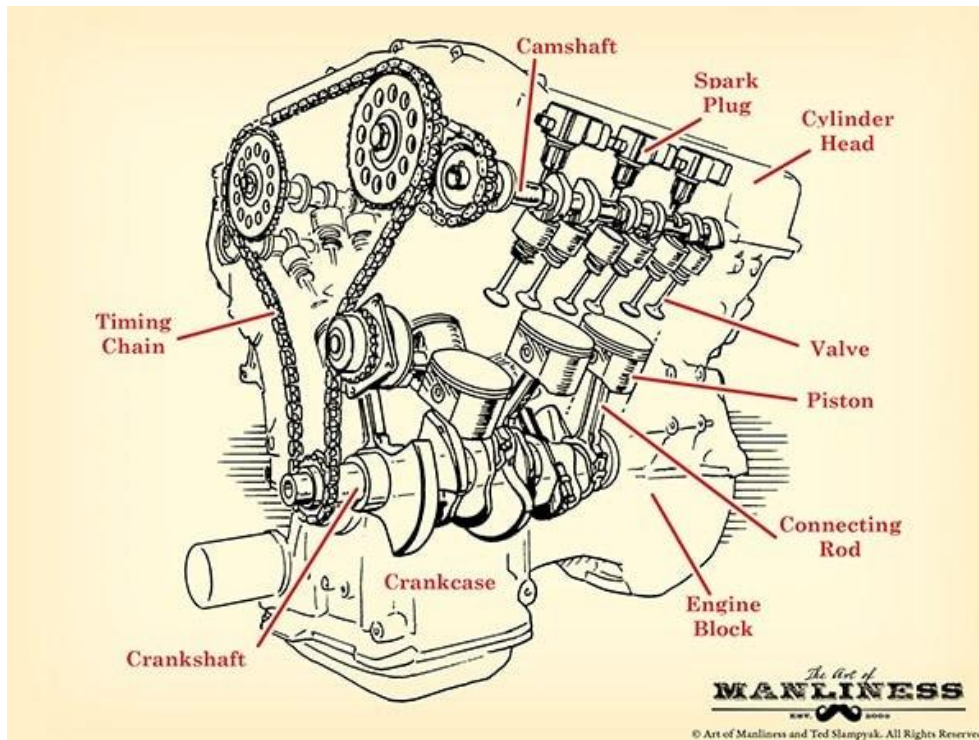
2. Podstawowe informacje o budowie silnika samochodowego

Silnik spalinowy tłokowy, stanowiący serce większości współczesnych samochodów, jest przykładem silnika wewnętrznego spalania (*ang. internal combustion engine*), w którym energia powstaje w wyniku spalania mieszanki paliwowo-powietrznej wewnątrz cylindrów. W przeciwieństwie do silników zewnętrznego spalania (np. parowych) proces ten zachodzi bezpośrednio w komorze spalania, co zwiększa sprawność energetyczną [2].

2.1. Kluczowe komponenty silnika

1. Blok silnika (*ang. engine block*) - stanowi podstawową konstrukcję, w której osadzone są cylindry. W silniku 6G74 Mistubishi Pajero (V6), na którym oparty jest wykorzystywany w aplikacji model, wykonany jest ze stopu aluminium, co redukuje masę, a cylindry ułożone są w układzie widlastym (kąt 60°)[1]. Każdy cylinder współpracuje z tłokiem, który przetwarza energię spalin na ruch posuwisto-zwrotny.
2. Głowica cylindra (*ang. cylinder head*) - montowana na bloku silnika, zawiera zawory (ssące i wydechowe), układ rozrządu oraz świece zapłonowe. W silniku 6G74 zastosowano hydrauliczne regulatory luzu, które automatycznie kompensują zużycie, eliminując konieczność regulacji zaworów[1][4]. Głowica połączona jest z blokiem za pomocą uszczelki, której prawidłowe dokręcenie jest kluczowe dla szczelności.
3. Tłoki i układ korbowy (*ang. piston* oraz *crankshaft*) - tłoki poruszają się w cylindrach, połączone z wałem korbowym za pomocą korbowodów. Wał korbowy przetwarza ruch liniowy tłoków na obrotowy, napędzając koła samochodu.
4. Układ rozrządu (*ang. camshaft*) - składa się z wałka rozrządu, paska rozrządu oraz zaworów. W silniku 6G74 zastosowano podwójny wałek rozrządu na bank cylindrów, co pozwala na precyzyjne sterowanie fazami otwarcia zaworów[1][4]. Synchronizacja wałka rozrządu z wałem korbowym jest zapewniona przez pasek rozrządu, którego napięcie regulowane jest przez automatyczny napinacz.

5. Układ smarowania - olej pompowany jest z miski olejowej do głównych kanałów smarujących, chłodzony przez wymiennik ciepła.

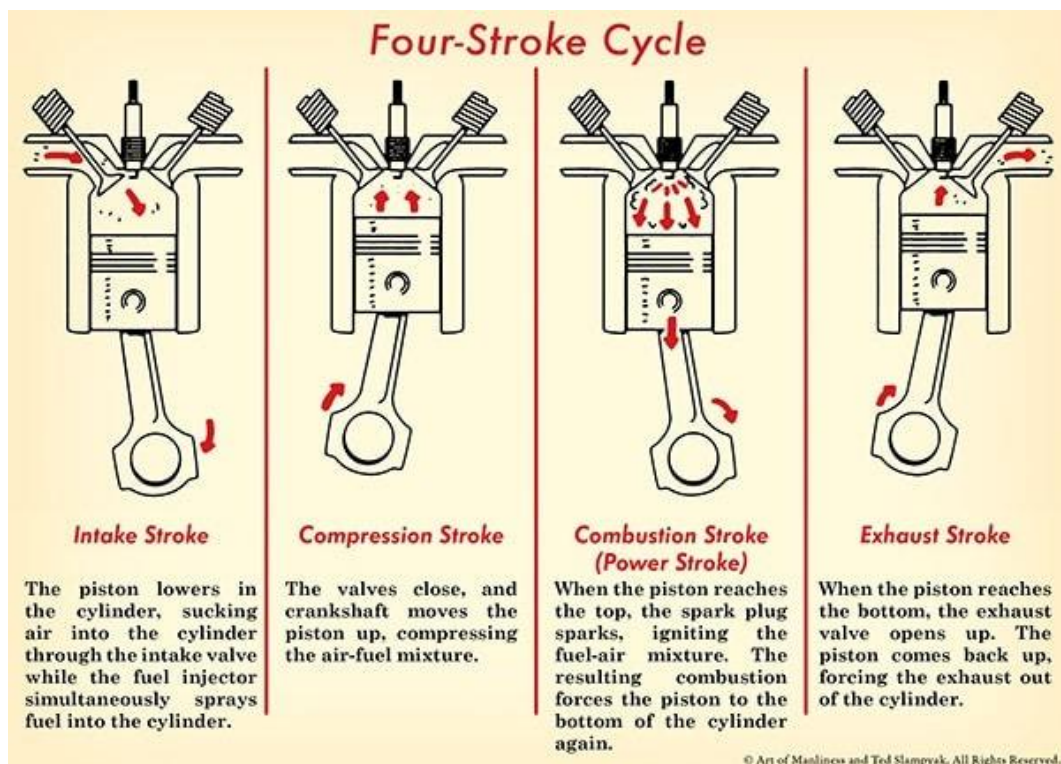


Rysunek 2.1 Wizualne przedstawienie budowy silnika [2].

2.2. Cykl czterosuwowy

Silnik 6G74 pracuje w cyklu czterosuwowym:

1. Ssanie (ang. *intake*) – zawór ssący otwiera się, tłok schodzi, zasysając mieszankę paliwową.
2. Sprężanie (ang. *compression*) – zawory zamknięte, tłok spręża mieszankę[2][5].
3. Praca (ang. *combustion*) – iskra zapala mieszankę, eksplozja przesuwa tłok w dół.
4. Wydech (ang. *exhaust*) – zawór wydechowy otwiera się, tłok wypycha spaliny.



Rysunek 2.2 Wizualne przedstawienie cyklu czterosuwowego [2].

3. Metodyka pracy oraz użyte urządzenia i oprogramowanie

W tej części pracy scharakteryzowano metodykę badań użytkowników, urządzenia wirtualnej rzeczywistości wykorzystane w projekcie, a także opisano oprogramowanie służące do stworzenia aplikacji VR.

3.1. Metodyka pracy

Badanie użytkownika zostało zaprojektowane w celu oceny efektywności aplikacji VR w kontekście szkolenia z montażu silnika samochodowego. Proces badawczy składa się z trzech głównych etapów: ankietowania wstępnego, symulacji w środowisku VR oraz analizy danych.

1. Ankieta wstępna

Przed rozpoczęciem symulacji użytkownik odpowiada na dwa pytania:

- "Jak oceniasz swoją wiedzę o motoryzacji w skali 1-10?" – pozwala to zaklasyfikować uczestnika do grupy o niskim/średnim/wysokim poziomie wiedzy.
- "Czy miałeś/aś styczność z wirtualną rzeczywistością?" – identyfikuje potencjalne trudności związane z adaptacją do interfejsu VR.

2. Symulacja w środowisku VR

Użytkownik otrzymuje gogle Meta Quest 2 i jest wprowadzany do aplikacji. Proces montażu silnika podzielony jest na sekwencyjne grupy zadaniowe, które odblokowywane są po kolei. Każda grupa składa się z części, a czas montażu poszczególnych elementów jest mierzony przez system.

3. Analiza danych

Po zakończeniu symulacji zbierane są:

- czasy cząstkowe dla każdej części
- czas całkowity montażu silnika
- dane z ankiety wstępnej – pozwalają skorelować wyniki z poziomem wiedzy i doświadczeniem w VR.

3.2. Wykorzystane urządzenia

Do realizacji projektu wybrano gogle VR Oculus Meta Quest 2 – samodzielne urządzenie, które nie wymaga podłączenia komputera, co umożliwiło

mobilne testy. Kluczową urzędzenia tego rozwiązania jest precyzyjne śledzenie ruchu kontrolerów, niezbędne do realistycznej interakcji z częściami silnika. Aplikacja została zoptymalizowana pod kątem ergonomii – krótki czas kalibracji oraz intuicyjne menu nawigacyjne pozwalają użytkownikom skupić się na zadaniu, a nie na obsłudze sprzętu.

3.3. Wykorzystane oprogramowanie

Podstawowym narzędziem do stworzenia aplikacji był silnik Unity w wersji 2022.3.25f1, wybrany ze względu na elastyczność w projektowaniu środowisk 3D i wsparcie w technologii VR[3]. Unity umożliwiło implementację logiki montażu poprzez komponentową architekturę – każda część silnika została zdefiniowana jako niezależny obiekt z własnymi skryptami w języku C#, kontrolującymi interakcje (np. wykrywanie kolizji z gniazdami). Intuicyjny edytor scen pozwolił na precyzyjne rozmieszczenie modeli 3D w przestrzeni wirtualnej.

Kluczowe modele 3D części silnika powstały w programie Blender, wykorzystywanym do modelowania i optymalizacji siatek trójkątów pod kątem wydajności w VR. Narzędzia oprogramowania Blender pozwoliły zachować szczegóły techniczne przy jednoczesnej redukcji liczby poligonów. Gotowe modele eksportowano do formatu FBX, a następnie importowano do Unity, gdzie dodawano materiały oraz komponenty fizyki. Integracja obu programów zapewniła spójność między realistycznym wyglądem części, a ich zachowaniem w aplikacji.

3.4. Zbieranie i przetwarzanie danych pomiarowych

Dane zbierane są w dwóch fazach:

1. Ankieta wstępna – przed rozpoczęciem symulacji użytkownik wprowadza:
 - Poziom wiedzy motoryzacyjnej w skali 1-10 (np. 1 – brak wiedzy, 10 – ekspert).
 - Doświadczenie z VR (tak/nie), aby zidentyfikować ewentualne trudności adaptacyjne.
2. Symulacja montażu w VR – system automatycznie rejestruje:
 - Czas montażu każdej części z dokładnością do 0.01s.

- Kolejność realizowanych grup, co pozwala analizować postęp użytkownika.

Dane są ręcznie wprowadzane do programu Excel i zapisywane w formacie CSV i analizowane z wykorzystaniem RStudio. Wykorzystywane są one do porównywania czasu montażu między użytkownikami o różnym poziomie wiedzy i analizy czy doświadczenie z VR wpływa na szybkość nauki.

4. Opis świata rzeczywistego

Ten rozdział omawia bardziej szczegółowo samą aplikację, stosowane technologie, funkcjonalności oraz działanie, które umożliwia wydajne wykonywanie planowanych celów.

4.1. Opis zasobów ludzkich

Aplikacja skierowana jest przede wszystkim do:

- Uczniów szkół technicznych i mechaników stażystów – umożliwia opanowanie sekwencji montażowych bez konieczności dostępu do fizycznych silników.
- Ośrodków szkoleniowych – stanowi alternatywę dla kosztownych prototypów, redukując wydatki na materiały eksploatacyjne.
- Entuzjastów motoryzacji – oferuje możliwość poznania budowy silnika w formie interaktywnego doświadczenia.

Intuicyjny interfejs i brak wymogu wcześniejszego doświadczenia z VR czy ze znajomością budowy silnika sprawiają, że aplikacja jest dostępna dla szerokiego grona użytkowników. Jedynym warunkiem jest podstawowa sprawność manualna w obsłudze kontrolerów.

4.2. Dane techniczne

By móc korzystać z aplikacji niezbędne są gogle VR Oculus Meta Quest 2 razem z dołączonymi do niego kontrolerami.

4.3. Wymagania funkcjonalne i нефunkcjonalne

Wymagania funkcjonalne:

- Aplikacja sekwencyjnie odblokowuje następne części.
- Aplikacja mierzy czas, który użytkownik potrzebuje na zamontowanie części.
- Aplikacja wyświetla w postaci tablicy czasy użytkownika poświęcone na poszczególne części.
- Aplikacja wyświetla dynamiczne instrukcje tekstowe informujące o następnych krokach.

Wymagania нефunkcjonalne:

- Aplikacja jest wydajna, zapewnia płynność renderowania powyżej 60 klatek na sekundę (docelową płynnością jest 90 klatek).
- Aplikacja jest prosta w obsłudze.
- Aplikacja wykorzystuje czytelny i intuicyjny interfejs.

5. Diagramy UML

5.1. Diagram przypadków użycia

W systemie wyróżniono jednego głównego aktora – Użytkownika, który realizuje wszystkie funkcjonalności systemu w środowisku wirtualnej rzeczywistości. Na rysunku 5.1 przedstawiono diagram przypadków użycia z przyporządkowaniem do aktora.

Przypadek użycia	Rozpocznij montaż
Aktor	Użytkownik
Warunki początkowe	Użytkownik znajduje się w menu głównym aplikacji.
Zdarzenie inicjujące	Użytkownik chce rozpocząć szkolenie.
Scenariusz podstawowy	Użytkownik wybiera z menu opcję „Start”. System odblokowuje pierwszą grupę części i przenosi użytkownika do sceny montażowej.
Rezultat	Rozpoczęcie procesu montażu z dostępną pierwszą grupą części.

Tabela 5.1 Przypadek użycia – Rozpocznij montaż.

Przypadek użycia	Odblokuj kolejną grupę części
Aktor	Użytkownik
Warunki początkowe	Użytkownik ukończył montaż poprzedniej grupy części.
Zdarzenie inicjujące	Użytkownik poprawnie zamontował aktualną grupę części.
Scenariusz podstawowy	System weryfikuje poprawność montażu i odblokowuje następną grupę części.
Rezultat	Dostęp do nowej grupy części do montażu.

Tabela 5.2 Przypadek użycia – Odblokuj kolejną grupę części.

Przypadek użycia	Wyświetl instrukcje dynamiczne
Aktor	Użytkownik
Warunki początkowe	Użytkownik jest w trakcie montażu grupy części.
Zdarzenie inicjujące	System odblokował nową grupę części.
Scenariusz podstawowy	System automatycznie wyświetla instrukcję dla aktualnej grupy części w panelu InstructionScreen.
Rezultat	Użytkownik widzi instrukcję montażu dla aktualnej grupy części.

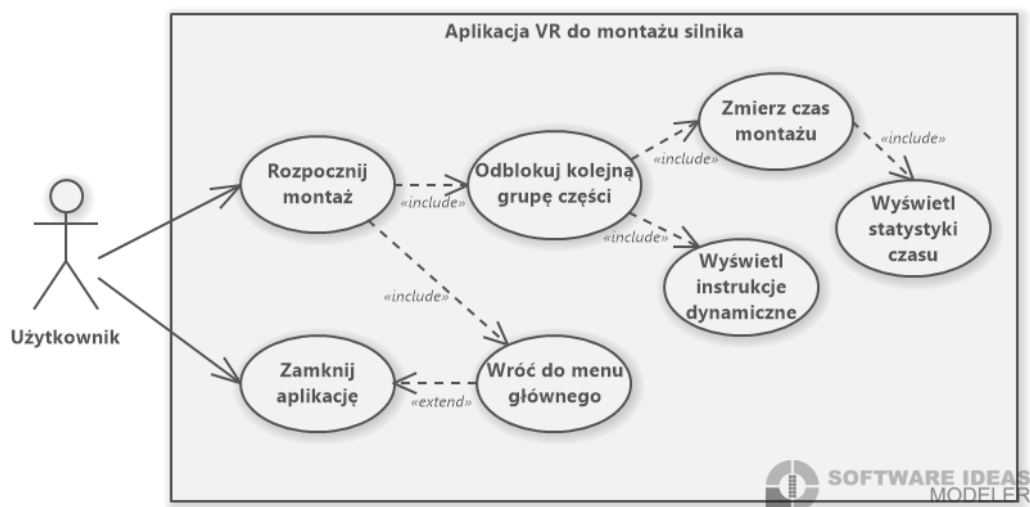
Tabela 5.3 Przypadek użycia – Wyświetl instrukcje dynamiczne.

Przypadek użycia	Zmierz czas montażu
Aktor	Użytkownik
Warunki początkowe	Użytkownik rozpoczął montaż grupy części.
Zdarzenie inicjujące	Użytkownik montuje pierwszą część w grupie.
Scenariusz podstawowy	System automatycznie rejestruje czas od momentu zamontowania poprzedniej części do umieszczenia następnej w gnieździe.
Rezultat	Czas montażu danej części zostaje zapisany w systemie.

Tabela 5.4 Przypadek użycia – Zmierz czas montażu.

Przypadek użycia	Wyświetl statystyki czasu
Aktor	Użytkownik
Warunki początkowe	Użytkownik ukończył montaż co najmniej jednej części.
Zdarzenie inicjujące	Użytkownik znajduje się w scenie montażowej.
Scenariusz podstawowy	System automatycznie aktualizuje panel wyświetlający czasy po każdym zamontowaniu części.
Rezultat	Statystyki czasu są stale widoczne w interfejsie bez konieczności interakcji.

Tabela 5.5 Przypadek użycia – Wyświetl statystyki czasu.

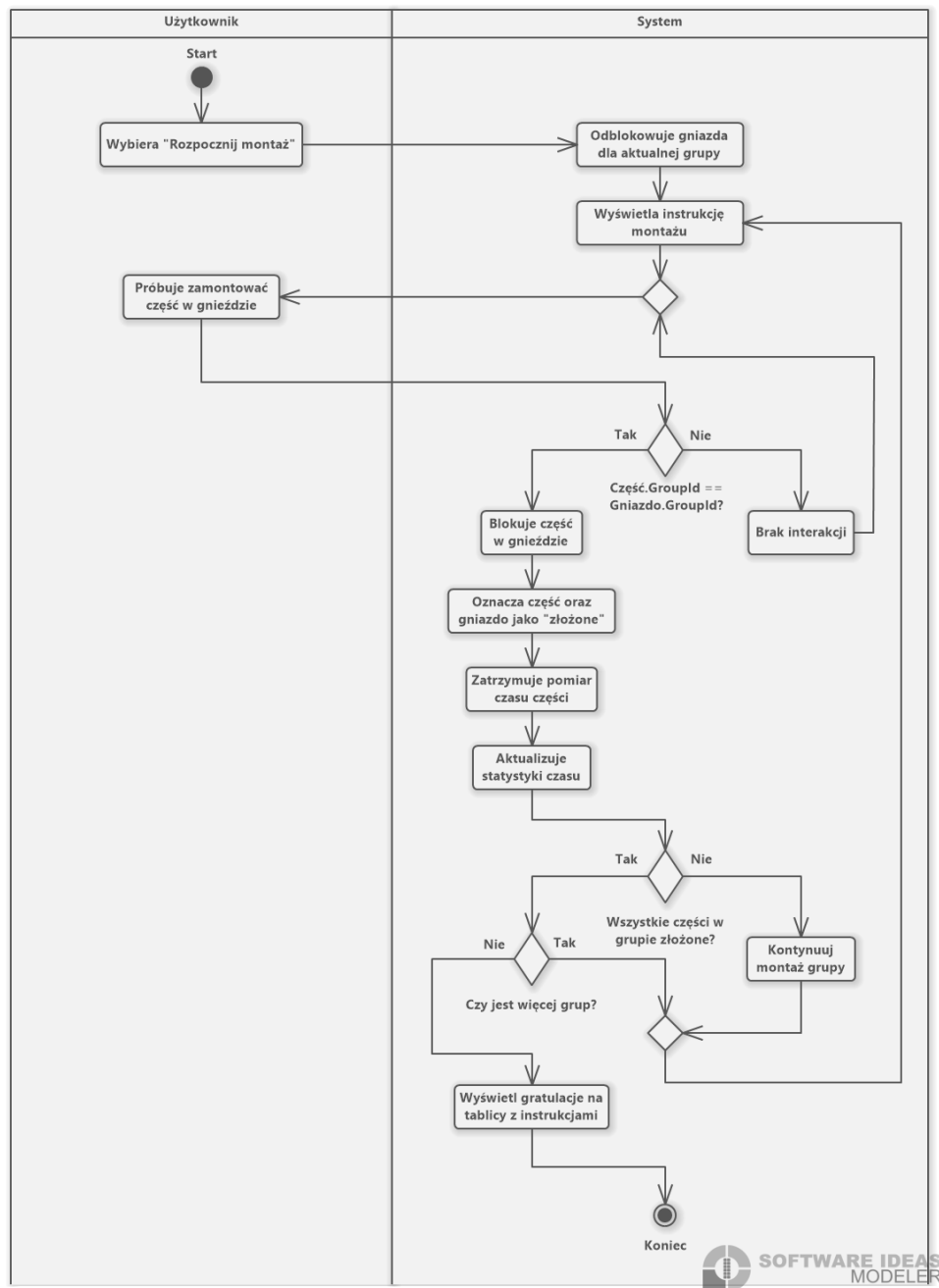


Rysunek 5.1 Diagram przypadków użycia.

5.2. Diagram aktywności

Na poniżej zamieszczonym rysunku 5.2 można znaleźć diagram aktywności montażu silnika. Proces rozpoczyna się od wybrania przez użytkownika w menu głównym opcji „Start”, co przenosi nas do sceny montażu i inicjuje odblokowanie pierwszej grupy części oraz wyświetlenie instrukcji montażu na tablicy. Użytkownik manipuluje fizycznie częściami, próbując umieścić je w odpowiednich

gniazdach. Jeśli prawidłowa część zostanie umieszczona w gnieździe to zostanie ona trwale zablokowana, a czas jaki był potrzebny na montaż zostaje zapisany i wyświetlony na tablicy. W przypadku próby umieszczenia nieprawidłowej części system uniemożliwia interakcję, wymuszając odnalezienie prawidłowej części. Po złożeniu wszystkich części w grupie odblokowywana zostaje kolejna z grup. W sytuacji kiedy wszystkie grupy zostały złożone zostaje wyświetlona krótka informacja gratulująca użytkownikowi złożenia wszystkich części na tablicy z instrukcjami.



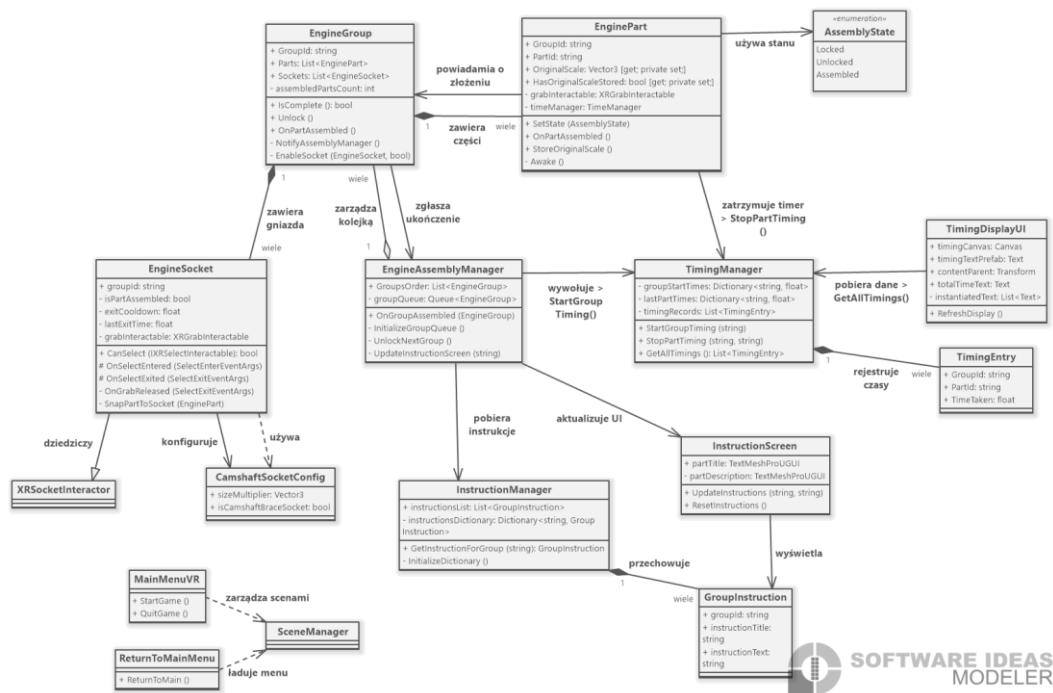
Rysunek 5.2 Diagram aktywności składania silnika.

5.3. Diagram klas

Rysunek 5.3. przedstawia diagram klas systemu. Głównym komponentem sterującym procesem jest `EngineAssemblyManager`, który zarządza kolejką grup montażowych (`EngineGroup`) w zdefiniowanej sekwencji. Każda grupa składa się z części (`EnginePart`) oraz dedykowanych gniazd montażowych (`EngineSocket`), których interakcje są kontrolowane przez maszynę stanów (`AssemblyState`). Użytkownik rozpoczyna proces od momentu wciśnięcia przycisku „Start” w menu głównym, co przenosi go do sceny montażu silnika, gdzie następuje odblokowanie pierwszej grupy i inicjalizuje pomiar czasu w `TimeManager` oraz aktualizację instrukcji w panelu `InstructionScreen`.

Kluczową rolę pełni mechanizm wykrywania poprawności montażu: gdy poprawna część zostaje umieszczona w gnieździe (`EngineSocket.CanSelect`), system wywołuje metodę `OnPartAssembled`, która aktualizuje licznik złożonych elementów danej grupy. Po skompletowaniu wszystkich części w grupie, `EngineAssemblyManager` odblokowuje następną sekwencję, jednocześnie zapisując czasy wykonania w strukturze `TimingEntry`. Dane te są wizualizowane w interfejsie `TimingDisplayUI` poprzez metodę `RefreshDisplay`.

Architektura wykorzystuje zasadę enkapsulacji – pola takie jak `assembledPartCount` w `EngineGroup` lub `timingRecords` w `TimeManager` są prywatne, a dostęp do nich odbywa się poprzez publiczne metody (np. `Unlock`, `StopPartTiming`). Integracja z silnikiem Unity jest widoczna w dziedziczeniu `EngineSocket` po `XRSocketInteractor` oraz wykorzystywaniu `SceneManager` do zarządzania scenami w menu głównym.



Rysunek 5.3 Diagram klas aplikacji.

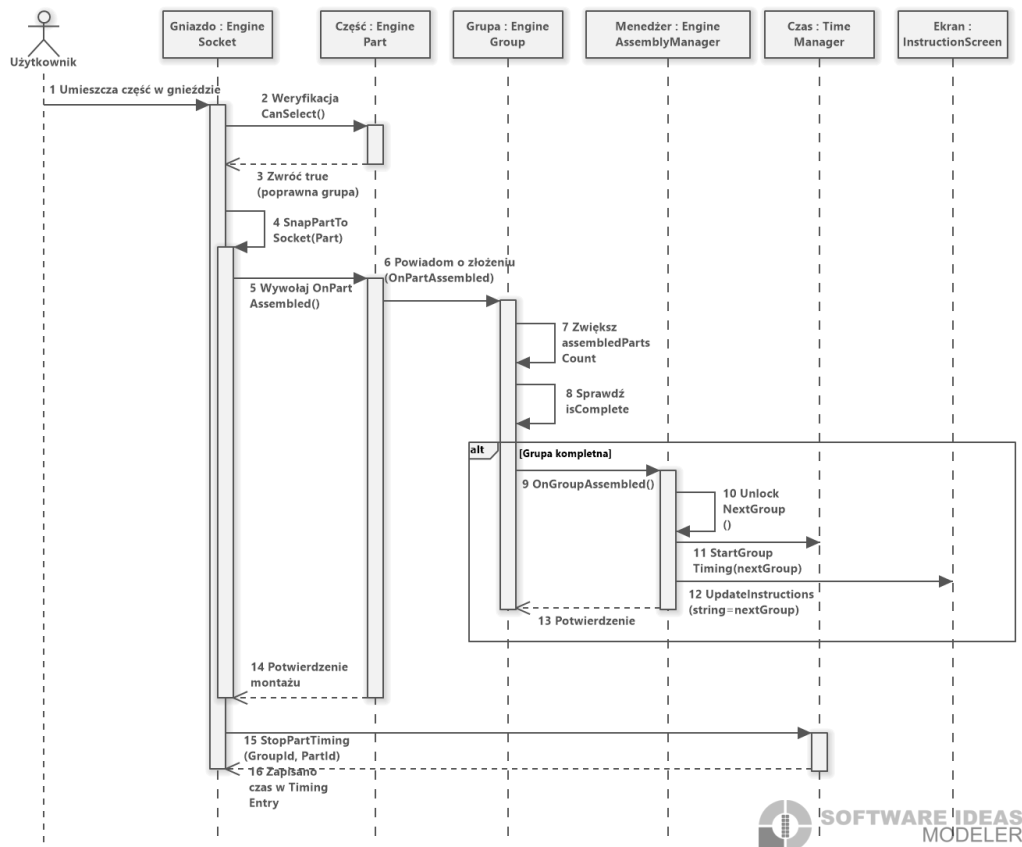
5.4. Diagram sekwencji

Na rysunku 5.4. można znaleźć przedstawienie sekwencji interakcji podczas poprawnego montażu części silnika. Proces inicjowany jest przez użytkownika, który fizycznie umieszcza wybraną część (EnginePart) w dedykowanym gnieździe montażowym (EngineSocket). System weryfikuje zgodność części z grupą (krok 2 i 3: CanSelect()), wykorzystując mechanizm warstw interakcji z XR Interaction Toolkit. Po pozytywnej walidacji następuje automatyczne pozycjonowanie części (krok 5: SnapPartToSocket), które skutkuje aktualizacją stanu logicznego poprzez wywołanie metody OnPartAssembled() (krok 6).

Kluczowym elementem jest komunikacja między komponentami:

- Część → Grupa (krok 7): powiadomienie o złożeniu wyzwała aktualizację licznika assembledPartsCount oraz sprawdzenie kompletności grupy.
- Grupa → Menedżer (krok 10): w przypadku kompletności, grupa inicjuje procedurę odblokowania następnej sekwencji montażowej poprzez EngineAssemblyManager.
- Integracja z podsystemami:
 - TimeManager rozpoczyna pomiar czasu dla nowej grupy
 - InstructionScreen aktualizuje wyświetlane instrukcje

Diagram podkreśla jednokierunkowy przepływ sterowania – część nie oczekuje odpowiedzi od grupy po zgłoszeniu złożenia (krok 7 → 14), co odzwierciedla zasadę niskiego sprzężenia w architekturze systemu. Mechanizm pomiaru czasu jest inicjowany dwufazowo: zatrzymanie mierzenia czasu dla obecnej części (krok 15) i uruchomienie pomiaru dla nowej grupy (krok 11), co zapewnia ciągłość zbierania metryk.



Rysunek 5.4 Diagram sekwencji montażu jednej z części.

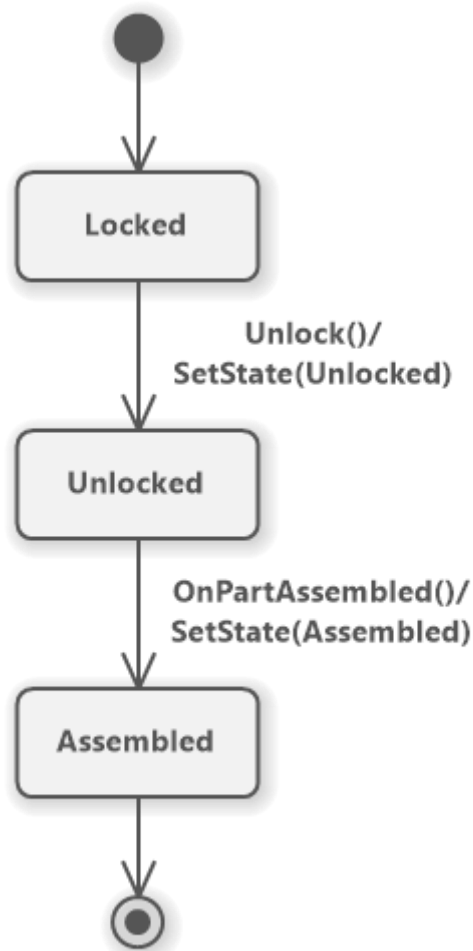
5.5. Diagram stanów

Rysunek 5.5 przedstawia diagram stanów dla komponentu EnginePart, ilustrujący trzy kluczowe stany związane z procesem montażu silnika.

Stan początkowy – Locked - charakteryzuje się możliwością podnoszenia części przez użytkownika, przy jednoczesnej blokadzie umieszczania jej w gniazdach montażowych. Mechanizm ten jest realizowany poprzez konfigurację warstw interakcji, które ograniczają możliwość łączenia z gniazdami. Część pozostaje widoczna, ale nie może być trwale zamontowana.

Stan `Unlocked` aktywuje pełną interakcję montażową. Przejście inicjowane jest metodą `Unlock()`, która aktualizuje warstwy kolizyjne (`socketInteractionLayers`), umożliwiając umieszczenie części w dedykowanych gniazdach. W tym stanie system rozpoczyna pomiar czasu montażu poprzez integrację z `TimeManager`.

Stan końcowy – `Assembled` – następuje po prawidłowym złożeniu części, wyłączając wszystkie formy interakcji. Część jest trwale unieruchamiana w gnieździe, co symuluje fizyczne zamocowanie elementu. Proces ten jest nieodwracalny bez wcześniejszego powrotu do menu głównego.



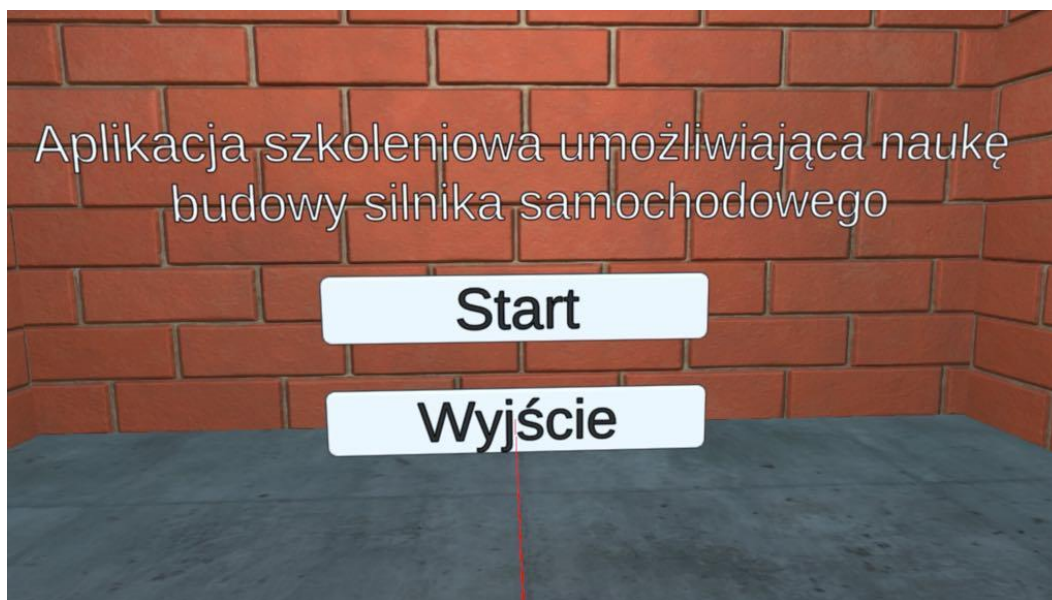
Rysunek 5.5 Diagram stanu komponentu `EnginePart`.

6. Interfejs użytkownika

W tym rozdziale zostanie w skrócie omówiony interfejs w oparciu o rzuty ekranu wykonane w VR w trakcie korzystania z aplikacji. Na tych rzutach ekranu ukazane zostaną podstawowe funkcjonalności oraz zastosowania wcześniej omawianych komponentów aplikacji.

Rysunek 6.1 przedstawia ekran główny aplikacji szkoleniowej. Po uruchomieniu systemu użytkownik zostaje umieszczony w wirtualnym pomieszczeniu, gdzie centralnym elementem jest panel sterujący zawieszony w przestrzeni 3D w ergonomicznej odległości 2-3 metrów od punktu obserwacyjnego. W panelu znajdują się dwa przyciski:

1. Przycisk „Start” inicjuje proces szkoleniowy, przenosząc użytkownika do interaktywnej sceny montażowej, gdzie możliwe jest rozpoczęcie nauki budowy silnika. Jego funkcjonalność skupia się na płynnym przejściu do głównego modułu aplikacji.
2. Przycisk „Wyjście” odpowiada za bezpieczne zamknięcie aplikacji, zapewniając zwolnienie zasobów systemowych i zakończenie sesji.

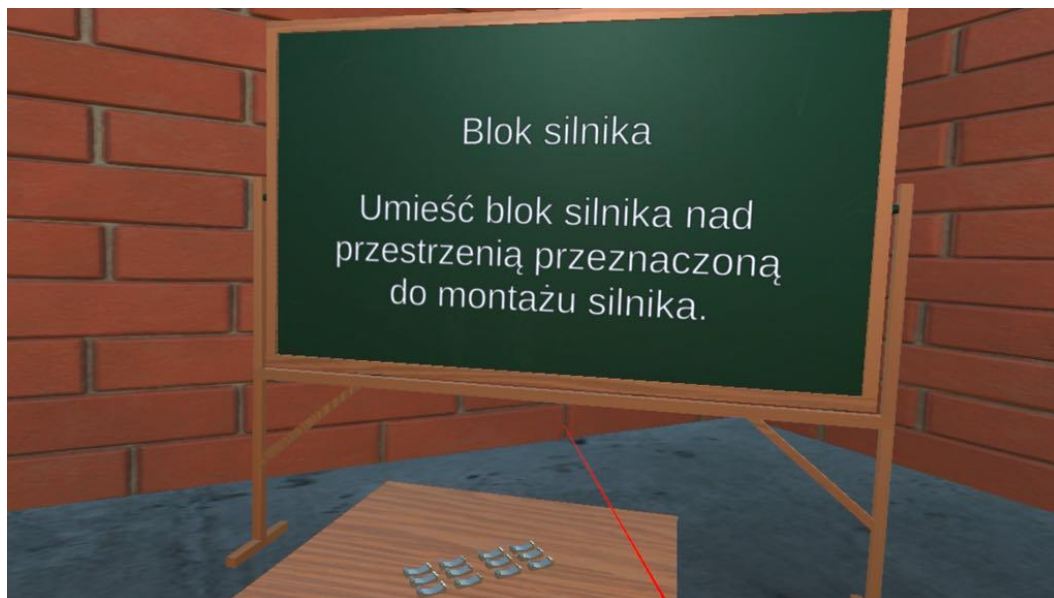


Rysunek 6.1 Menu główne.

Rysunek 6.2 przedstawia interfejs instrukcji montażowych. Główny panel informacyjny umieszczony jest w strategicznym miejscu sceny, zapewniając natychmiastową widoczność po załadowaniu przestrzeni. Składa się z dwóch kluczowych sekcji:

1. Nazwa części: Wyświetla nazwę aktualnej części.
2. Instrukcja: Zawiera skrócony opis czynności do wykonania.

Panel dynamicznie aktualizuje treść w zależności od postępu użytkownika, współpracując z systemem śledzenia stanu części.



Rysunek 6.2 Tablica z instrukcjami.

Rysunki 6.3 i 6.4 przedstawiają przestrzenną organizację komponentów silnika w środowisku szkoleniowym. Części rozmieszczone są na stołach wokół użytkownika w kolejności w jakiej powinny być montowane, tworząc logiczny układ, który wspomaga proces nauki. Dodatkowo na rysunku można zaobserwować na podłodze pomieszczenia wydzielone kwadratowe przestrzenie - jest to system teleportacji umożliwiający poruszanie się po przestrzeni szkoleniowej. Punkty umożliwiające teleportacje znajdują się przy stołach z częściami, wokół przestrzeni montażu silnika oraz przed tablicą czasów dając możliwość na spojrzenie na nią z komfortowej pozycji.

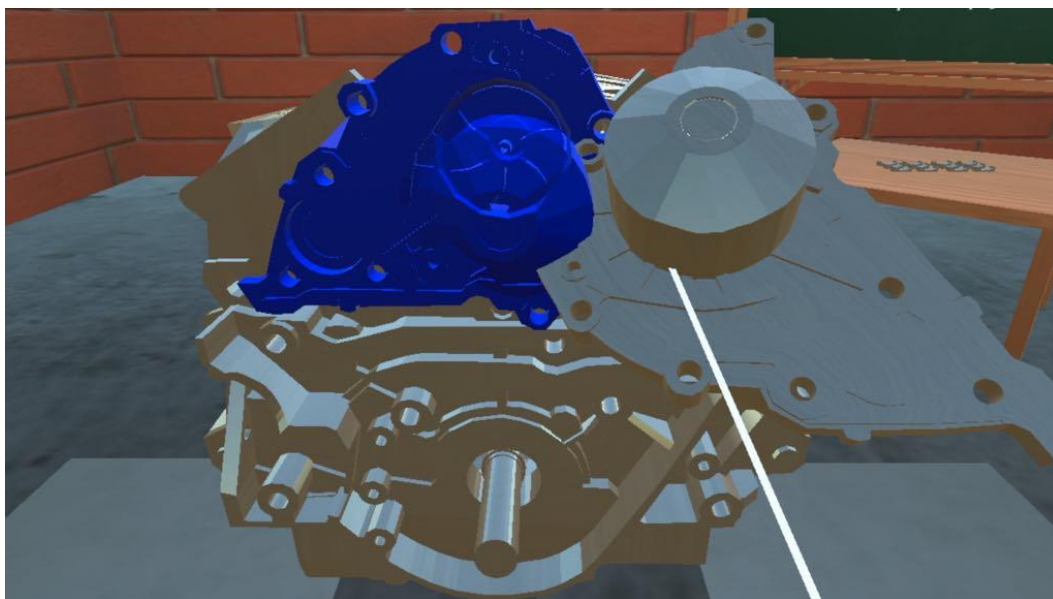


Rysunek 6.3 Pierwszy stół z częściami.



Rysunek 6.4 Drugi stół z częściami oraz widoczne teleporty.

Rysunek 6.5 prezentuje mechanizm wizualnego wsparcia montażu, wykorzystujący dynamiczną poświatę do precyzyjnego pozycjonowania części. Gdy użytkownik zbliża część do dedykowanego gniazda, system wyświetla niebieską poświatę w przestrzeni obszaru docelowego.



Rysunek 6.5 Poświata ukazująca przestrzeń montażu.

Rysunek 6.6 przedstawia interaktywną tablicę wyników czasowych, zlokalizowaną na ścianie głównej przestrzeni montażowej. Tablice umożliwia sprawdzenie czasów jakie użytkownik potrzebował na wykonanie szkolenia i złożenie silnika w całości z podziałem na poszczególne części. Aby móc zobaczyć wszystkie swoje czasy użytkownik ma możliwość poruszania tablicą w trybie góra-dół. Poza czasami częściowymi można również na tablicy znaleźć sumę czasu użytkownika oraz dodatkowo przycisk powrotu do menu głównego.



Rysunek 6.6 Tablica czasów.

7. Analiza i omówienie wyników pracy

Badanie miało na celu ocenę efektywności aplikacji VR w nauce montażu silnika samochodowego. W analizie wykorzystano dane 15 użytkowników, którzy ułożyli 34 części na swoich miejscach. Poniżej przedstawiono kluczowe etapy badania, metody statystyczne oraz wnioski.

7.1. Metodologia analizy

Dane zapisano w formacie CSV, zawierającym:

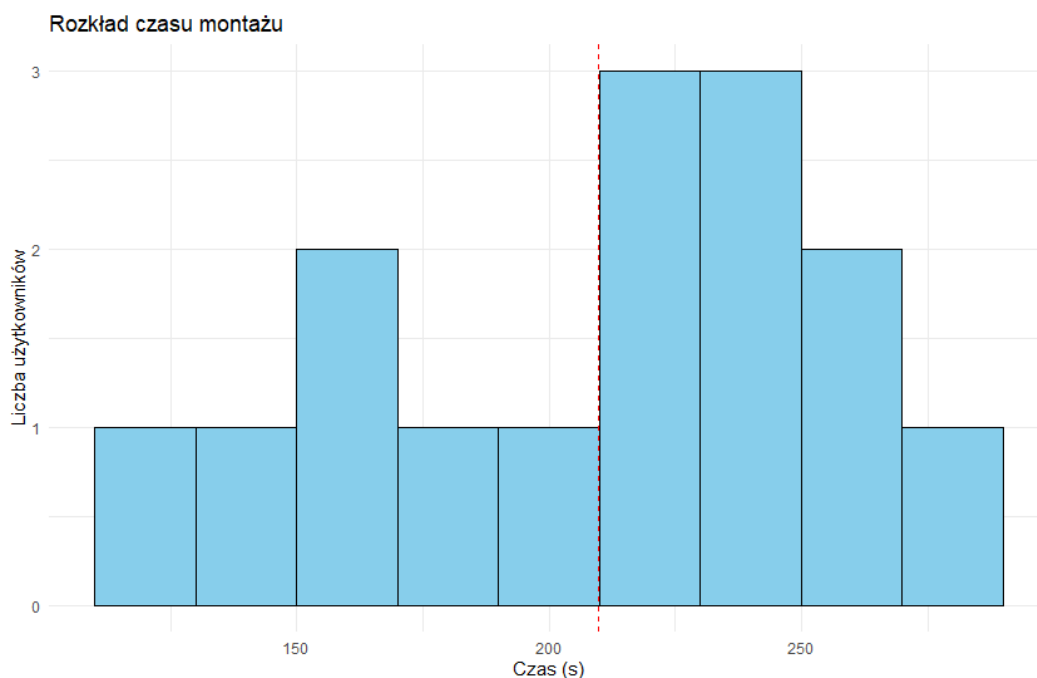
- Zmienne niezależne:
 - MotoKnowledge (1-10) – samoocena wiedzy motoryzacyjnej,
 - VRExperience (Tak/Nie) – doświadczenie z technologią VR.
- Zmienne zależne: Czas montażu 34 części silnika (w sekundach).

Analizę przeprowadzono w środowisku R 4.3.1 z wykorzystaniem pakietów:

- tidyverse – przetwarzanie danych,
- ggplot2 – wizualizacje,
- cluster – analiza skupień[8].

7.2. Statystyki opisowe

Średni czas montażu całego silnika wyniósł 209.79s. Najszybszy użytkownik uzyskał czas 123.81s, najwolniejszy 281.47s. Rozkład doświadczenia VR – 46.7% użytkowników miało wcześniejszy kontakt z technologią VR. Rozkład czasu montażu (Rysunek 7.1) wykazuje lekką asymetrię prawostronną, co sugeruje, że większość użytkowników mieściła się w przedziale 150–220 s. Użytkownicy z czasem >250 s (n=3) stanowili wyraźną grupę odstającą, co może wskazywać na potrzebę dodatkowego szkolenia dla osób o niskiej wiedzy motoryzacyjnej (średnia wiedza w tej grupie: 2/10).

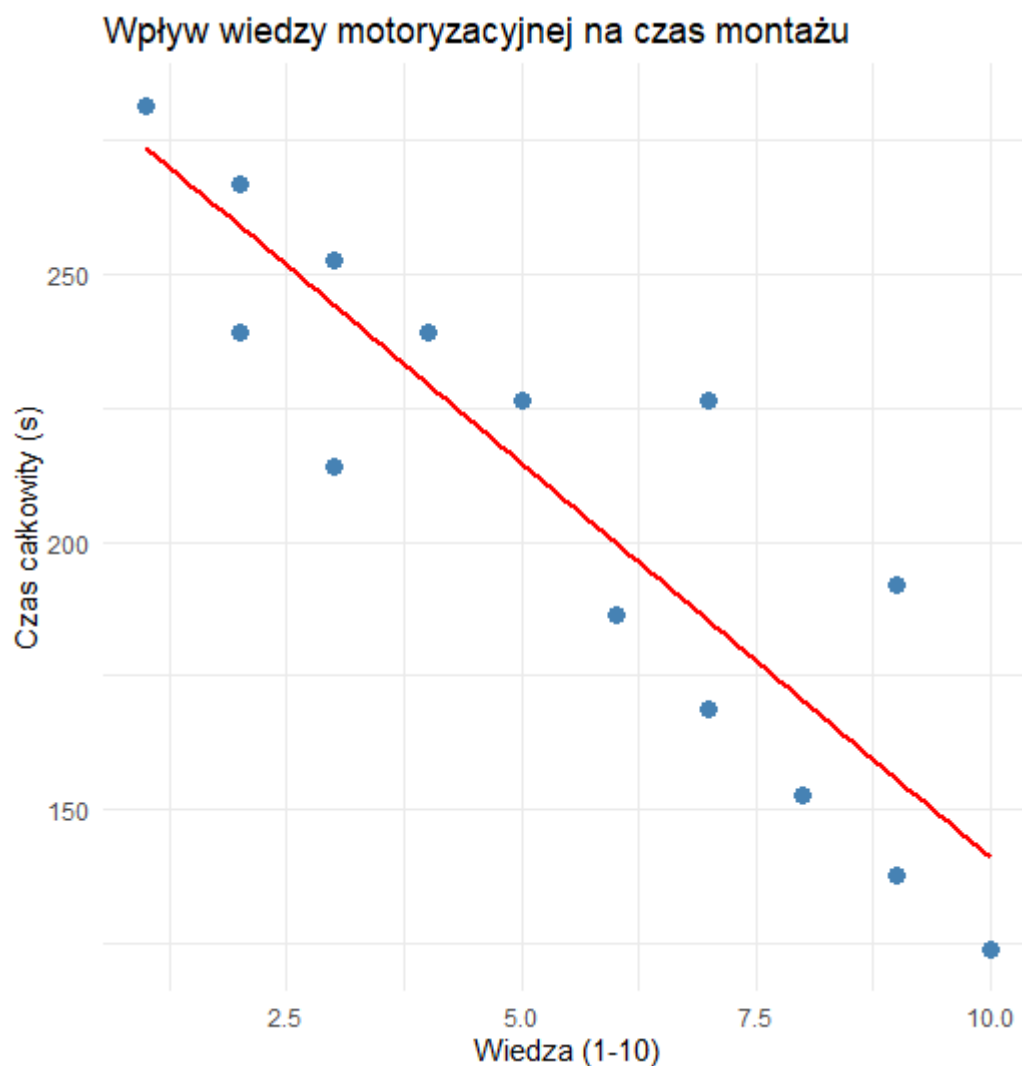


Rysunek 7.1 Rozkład czasu montażu wśród wszystkich użytkowników.

7.3. Wpływ wiedzy motoryzacyjnej

Analiza regresji liniowej wykazała silny ujemny związek między wiedzą motoryzacyjną a czasem montażu silnika ($\beta = -14.8$ s/punkt, $p < 0.001$)[9]. Każdy dodatkowy punkt w skali 1–10 skracał czas montażu średnio o 14.8 sekund, co przekłada się na różnicę 133 sekundy między użytkownikiem z wiedzą=10, a wiedzą=1 (Rysunek 7.1). Model wyjaśniał 89.6% wariancji czasu montażu ($R^2 = 0.804$), co wskazuje na jego wysoką wyjaśniającą moc.

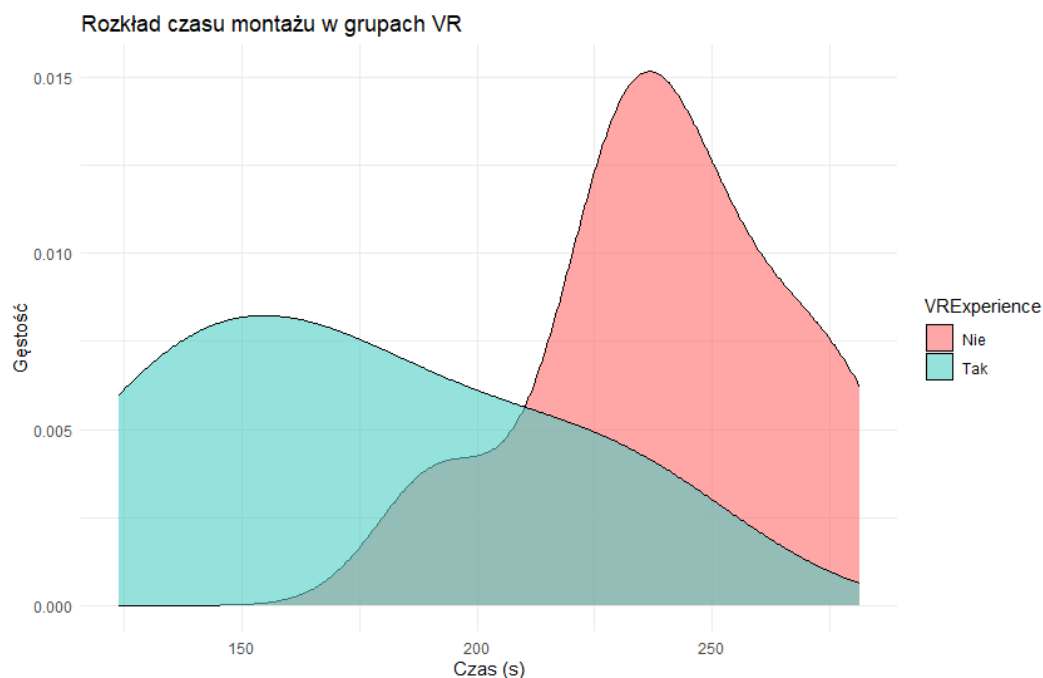
Wynik testu $p = 5.9e-06$ ($p < 0.001$) potwierdza wysoce istotny statystycznie charakter tej zależności. Diagnostyka residuów wykazała losowy rozkład błędów, co wspiera założenie liniowości modelu. Jedynym odstępstwem był użytkownik z wiedzą=7, który osiągnął czas 173 s – bliższy ekspertom niż średniej dla jego poziomu wiedzy, co sugeruje wpływ dodatkowych czynników (np. doświadczenia w VR, osobiste oceny wiedzy niższej w skali niż rzeczywiście mogłaby być zakwalifikowana).



Rysunek 7.2 Wiedza motoryzacyjna, a czas montażu.

7.4. Efekt doświadczenia VR

Efekt wcześniejszego doświadczenia z VR jest nie tylko statystycznie istotny ($p = 0.005$), ale także silny w praktyce ($d \text{ Cohen} = 1.38$). Użytkownicy z doświadczeniem osiągnęli czas montażu o 27.4% krótszy niż grupa bez doświadczenia, przy czym ich wyniki były bardziej spójne ($SD = 27.4s$ vs $SD = 41.6s$). Najlepszy wynik w grupie bez doświadczenia (192.1 s) był gorszy niż średnia dla grupy z doświadczeniem (174.6 s).



Rysunek 7.3 Doświadczenie z VR, a czas montażu.

7.5. Analiza skupień użytkowników

Metoda k-średnich wydzieliła trzy grupy użytkowników (Rysunek 7.3.), uwzględniając zarówno wiedzę motoryzacyjną, jak i czas montażu co jest zgodne z wcześniejszymi badaniami nad VR w edukacji technicznej[6]:

1. Eksperci (n=5):

- Średni czas montażu: 155.2s (zakres 135-173s),
- Średnia wiedza: 8,6/10,
- Charakterystyka: Najwyższa efektywność – połączenie wysokiej wiedzy i najkrótszego czasu. Grupa ta obejmuje zarówno użytkowników z wiedzą > 8, jak i jednego z wiedzą = 7, który wyróżnił się wyjątkowo niskim czasem montażu (168.7s).

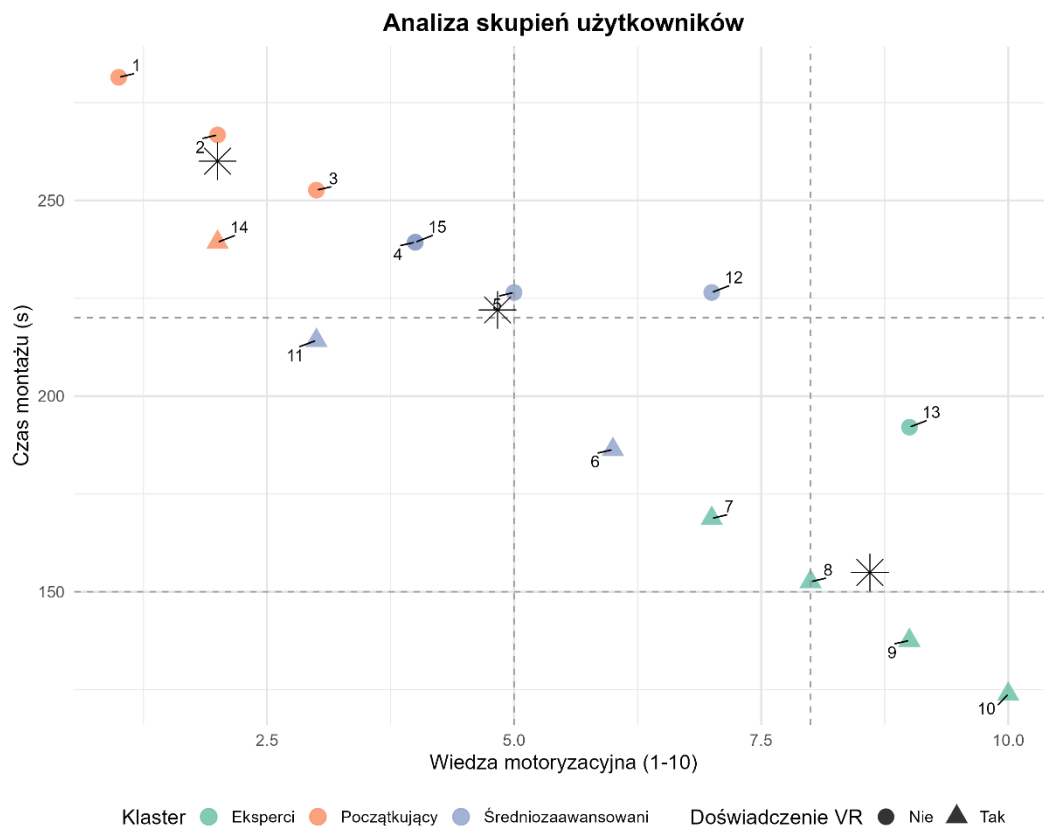
2. Średniozaawansowani (n=6):

- Średni czas: 222.3 s (zakres: 195–240 s),
- Średnia wiedza: 4.8/10,
- Charakterystyka: Umiarkowana wiedza i czas montażu. W grupie dominują użytkownicy bez doświadczenia z VR (83%).

3. Początkujący (n=4):

- Średni czas: 260.1 s (zakres: 245–278 s),

- Średnia wiedza: 2.0/10,
- Charakterystyka: Najdłuższy czas montażu i najniższa wiedza. Wszyscy użytkownicy w tej grupie nie mieli wcześniejszego kontaktu z VR.



Rysunek 7.4 Analiza skupień użytkowników.

8. Wnioski końcowe

Badanie wskazuje na potencjalną przydatność aplikacji VR w procesie nauki montażu silnika samochodowego, przy czym zaobserwowano korelację między czasem wykonania zadania, a dwoma czynnikami: poziomem wiedzy motoryzacyjnej oraz ich wcześniejszym doświadczeniem z technologiami VR.

1. Wiedza motoryzacyjna jest fundamentem efektywności

Analiza regresji wykazała, że każdy dodatkowy punkt wiedzy skraca czas montażu średnio o 14.8 s ($\beta = -14.8$, $p < 0.001$). Różnica między ekspertem (wiedza=10) a początkującym (wiedza=1) wynosi 133 s – to niemal 2.5-krotne przyspieszenie tempa pracy. Wynik ten podkreśla, że tradycyjna edukacja techniczna pozostaje niezbędna, nawet w kontekście szkoleń VR.

2. VR kompensuje braki w wiedzy

Użytkownicy z doświadczeniem VR montowali silnik 27.4% szybciej niż osoby bez takiego doświadczenia (d Cohena = 1.38). Co istotne, użytkownik z wiedzą=7, ale z VR, osiągnął czas 168.7s – porównywalny z ekspertami. Sugeruje to, że VR może niwelować deficyty wiedzy teoretycznej poprzez immersyjne symulacje praktyczne.

3. Trzy profile użytkowników wymagają różnego podejścia

Klasteryzacja wyodrębniła grupy: ekspertów, średniozaawansowanych i początkujących:

- Eksperti (n=5): Wymagają zaawansowanych scenariuszy (np. diagnozowanie usterek),
- Średniozaawansowani (n=6): Skorzystają z modułów powtórkowych i feedbacku w czasie rzeczywistym,
- Początkujący (n=4): Potrzebują podstawowych szkoleń z animacjami krok-po-kroku.

Aplikacje VR stanowią przełomowe narzędzie w edukacji motoryzacyjnej, co potwierdzają zarówno wyniki badań PwC [10], jak i praktyczne zastosowania w przemyśle, takie jak szkolenia techników Forda [7]. Wyniki niniejszej pracy są spójne z ogólnymi trendami wdrożeń VR w sektorze motoryzacyjnym [6]. Największą zaletą aplikacji jest demokratyzacja dostępu do specjalistycznych

szkoleń – użytkownicy z małych warsztatów czy regionów o ograniczonym dostępie do fizycznych silników mogą rozwijać kompetencje na równi z pracownikami dużych ośrodków. Kluczem do sukcesu jest jednak hybrydowy model szkoleniowy, łączący VR z tradycyjnymi metodami nauczania teorii.

9. Udoskonalenia aplikacji

W poniższym rozdziale przedstawiono propozycje udoskonaleń aplikacji, których celem jest podniesienie jakości jej funkcjonalności oraz wzbogacenie doświadczeń użytkowników. Zaproponowane zmiany obejmują zarówno rozbudowę mechanizmów interaktywnych, jak i wdrożenie innowacyjnych modułów wspierających proces szkoleniowy.

1. Integracja z serwerem

Wdrożenie systemu przesyłania danych na serwer w czasie rzeczywistym zamiast lokalnego przechowywania na tablicy wyników. Każdy czas montażu części oraz statystyki użytkowników byłyby automatycznie synchronizowane z bazą danych. Pozwoliłoby to na monitorowanie postępów użytkowników zdalnie, dane byłyby zabezpieczone przed utratą oraz umożliwiłoby to analizę wyników na dużych próbach.

2. Zaawansowana walidacja pozycjonowania części

Obecny mechanizm akceptuje część, gdy ta znajdzie się w gnieździe, niezależnie od precyzji. Wprowadzenie zwiększonej dokładności umożliwiłoby sprawdzanie np. ustawienia kąta czy zazębienia z sąsiednimi komponentami.

3. Tryb diagnozowania i naprawy usterek

Rozszerzenie aplikacji o kilka scenariuszy awaryjnych np.

- Symulacja zużytych części – użytkownik miałby za zadanie wymienienie jednej z części.
- Tryb diagnostyki – użytkownik miałby krótki opis przypadku i jego zadaniem by było znalezienie usterki oraz naprawienie jej. Umożliwiłoby to szkolenie nie tylko z samej budowy silnika, ale również umiejętności diagnostyki problemów i rozwiązywania ich.

4. Personalizowane poziomy trudności

Dodanie opcji wyboru poziomu trudności przed rozpoczęciem szkolenia. Poziom podstawowy dający instrukcje krok po kroku oraz poziom ekspercki z brakiem instrukcji i ograniczonym czasem na montaż.

10. Bibliografia

- [1] Mitsubishi Motors. *6G7 Engine Service Manual*. Mitsubishi Motors Corporation, 2001.
- [2] McKay, B. *How a Car's Engine Works*. The Art of Manliness, 2015.
- [3] Unity Technologies. *XR Interaction Toolkit Documentation*. Unity, 2023.
- [4] Mitsubishi-Forums.com. *Technical Discussions on 6G74 Engine*. Mitsubishi Enthusiast Forum.
- [5] Heywood, J. B. *Internal Combustion Engine Fundamentals*. McGraw-Hill, 1988.
- [6] G. Lan, Q. Lai, B. Bai, Z. Zhao and Q. Hao. *Virtual Reality for Automotive Engines Assembly and Disassembly*. IEEE, 2024.
- [7] Dearborn M. *How Ford, Bosch Are Using Virtual Reality to Train Technicians On All-Electric Mustang Mach-E*. Ford Newsroom, 2020.
- [8] Kavlakoglu E., Winland V. *What is k-means clustering*. IBM, 2020.
- [9] StatSoft. *Linear Regression Analysis*. 2024.
- [10] PwC. *What does virtual reality and the metaverse mean for training*. 2022.

11. Spis rysunków

Rysunek 2.1 Wizualne przedstawienie budowy silnika [2].	10
Rysunek 2.2 Wizualne przedstawienie cyklu czterosuwowego [2].....	11
Rysunek 5.1 Diagram przypadków użycia.	18
Rysunek 5.2 Diagram aktywności składania silnika.	19
Rysunek 5.3 Diagram klas aplikacji.	21
Rysunek 5.4 Diagram sekwencji montażu jednej z części.	22
Rysunek 5.5 Diagram stanu komponentu EnginePart	23
Rysunek 6.1 Menu główne.....	24
Rysunek 6.2 Tablica z instrukcjami.....	25
Rysunek 6.3 Pierwszy stół z częściami.	26
Rysunek 6.4 Drugi stół z częściami oraz widoczne teleporty.	26
Rysunek 6.5 Poświata ukazująca przestrzeń montażu.	27
Rysunek 6.6 Tablica czasów.....	27
Rysunek 7.1 Rozkład czasu montażu wśród wszystkich użytkowników.	29
Rysunek 7.2 Wiedza motoryzacyjna, a czas montażu.	30
Rysunek 7.3 Doświadczenie z VR, a czas montażu.....	31
Rysunek 7.4 Analiza skupień użytkowników.	32

12. Spis tabel

Tabela 5.1 Przypadek użycia – Rozpocznij montaż.	17
Tabela 5.2 Przypadek użycia – Odblokuj kolejną grupę części.....	17
Tabela 5.3 Przypadek użycia – Wyświetl instrukcje dynamiczne.....	17
Tabela 5.4 Przypadek użycia – Zmierz czas montażu.....	18
Tabela 5.5 Przypadek użycia – Wyświetl statystyki czasu.....	18

13. Streszczenie pracy

Niniejsza praca inżynierska poświęcona jest opracowaniu aplikacji szkoleniowej wykorzystującej technologię wirtualnej rzeczywistości (VR) do nauki budowy i montażu silnika samochodowego. Głównym celem projektu było stworzenie interaktywnego narzędzia, które umożliwia bezpieczne i efektywne szkolenie mechaników oraz entuzjastów motoryzacji, eliminując koszty i ograniczenia związane z tradycyjnymi metodami opartymi na fizycznych prototypach. Aplikacja, oparta na silniku Unity oraz technologii XR Interaction Toolkit, odwzorowuje proces montażu silnika Mitsubishi serii 6G7 z wykorzystaniem autorskich modeli 3D opracowanych w programie Blender. Kluczowe funkcjonalności obejmują sekwencyjne odblokowywanie grup gniazd, inteligentną walidację poprawności montażu, dynamiczne instrukcje tekstowe oraz system pomiaru czasu złożenia poszczególnych komponentów.

Badanie efektywności aplikacji przeprowadzono z udziałem 15 użytkowników, uwzględniając ich poziom wiedzy motoryzacyjnej i doświadczenie z technologią VR. Analiza statystyczna wykazała silną korelację między samooceną wiedzy uczestników a czasem montażu – każdy dodatkowy punkt w skali 1–10 skracał średni czas o 14,8 sekundy. Użytkownicy z wcześniejszym doświadczeniem w VR montowali silnik o 27% szybciej niż osoby bez takiego doświadczenia, co podkreśla znaczenie adaptacji do interfejsów immersyjnych. Klasteryzacja wyników wyodrębniła trzy grupy użytkowników: ekspertów (średni czas 155 s), średniozaawansowanych (222 s) i początkujących (260 s), co wskazuje na potrzebę zróżnicowanych ścieżek szkoleniowych.

Wnioski z pracy potwierdzają, że aplikacja VR stanowi wartościowe uzupełnienie tradycyjnych metod edukacji technicznej, szczególnie w kontekście redukcji kosztów i zwiększenia bezpieczeństwa szkoleń. Jednocześnie badania wykazały, że efektywność rozwiązania zależy od synergii między wiedzą merytoryczną a kompetencjami cyfrowymi użytkowników. Zaproponowano dalsze udoskonalenia systemu, takie integracja z chmurą danych, zaawansowana walidacja precyzji montażu oraz moduły diagnostyki usterek.

Opracowane narzędzie otwiera perspektywy dla standaryzacji szkoleń motoryzacyjnych i upowszechniania specjalistycznej wiedzy w branży.

14. Abstract

This engineering thesis presents the development of a virtual reality (VR)-based training application for learning the structure and assembly of a car engine. The primary objective of the project was to create an interactive tool that enables safe and cost-effective training for mechanics and automotive enthusiasts, eliminating the limitations of traditional methods reliant on physical prototypes. The application, built using the Unity engine and XR Interaction Toolkit, simulates the assembly process of a Mitsubishi 6G7 series engine, utilizing custom 3D models designed in Blender. Key features include sequential unlocking of engine part groups, intelligent real-time assembly validation, dynamic text-based instructions, and a time-tracking system for monitoring component assembly efficiency.

A study involving 15 users evaluated the application's effectiveness, considering participant's self-assessed automotive knowledge and prior VR experience. Statistical analysis revealed a strong correlation between user's technical expertise and assembly speed: each additional point on a 1–10 knowledge scale reduced average assembly time by 14.8 seconds. Participants with VR experience completed tasks 27% faster than those without, highlighting the importance of immersive interface familiarity. Cluster analysis identified three user groups: experts (average time: 155 s), intermediate users (222 s), and beginners (260 s), underscoring the need for tailored training approaches.

The results confirm that VR applications are a valuable supplement to traditional technical education, particularly in reducing costs and enhancing safety. Future enhancements proposed include cloud-based data integration, advanced precision validation, and fault diagnosis modules. This tool demonstrates significant potential for standardizing automotive training and broadening access to specialized knowledge in the industry.