

Specyfikacja Aplikacji Kursanta i aplikacji Trenera

Spis treści

1. Wstęp.....	2
2. Dokumentacja dla budowy docelowego systemu IT	2
2.1. Aplikacja trenera	2
2.1.1. Zarządzanie scenariuszami i modułami	2
2.1.2. Komunikacja z kursantem	2
2.1.3. Podgląd widoku kursanta	2
2.1.4. Wyświetlanie testowych informacji	3
2.1.5. Zarządzanie niektórymi ustawieniami aplikacji kursanta.....	3
2.2. Aplikacja kursanta	3
2.2.1. Budowa modułowa	3
2.2.2. Systemy interakcji	3
2.2.3. Implementacja interakcji z pilotem.....	3
2.2.4. Wizualizacja obiektów	4
2.2.5. Elementy środowiska	4
2.3. Opis wybranych elementów implementacji systemu IT	5
2.3.1. Komunikacja aplikacji z pilotem.....	5
2.3.2. System kierowania wzroku kursanta	5
2.3.3. Aktualizacja danych w aplikacji trenera.....	6
2.3.4. Warunki kroków i system celów.....	6
2.3.5. Dynamiczne ładowanie tekstów i zdjęć/video zadań	7
3. Załączniki	7

1. Wstęp

Docelowy system IT użyty do budowy aplikacji składa się z dwóch aplikacji: aplikacji przeznaczonej dla kursanta i aplikacji przeznaczonej dla trenera. Każda z nich działa w połączeniu z drugą, łącząc się na danym komputerze pozwalając na operacje w czasie trwania kursu. Aplikacja kursanta może również funkcjonować samodzielnie, ale pozbawiona będzie funkcjonalności dostarczanych przez aplikację trenera, jak np. możliwość wyboru modułu i scenariusza.

2. Dokumentacja dla budowy docelowego systemu IT

2.1. Aplikacja trenera

Aplikacja trenera pozwala na kontrolę kursu w czasie jego trwania. Zarządzanie modułami i wysyłanie wiadomości do użytkownika pozwala na lepsze przeprowadzenie kursu i szybszą reakcję na problemy.

Aplikacja trenera składa się z elementów takich jak:

- Zarządzanie scenariuszami i modułami,
- Komunikacja z kursantem
- Podgląd widoku kursanta
- Wyświetlenie testowych informacji aplikacji kursanta
- Zarządzanie niektórymi ustawieniami aplikacji kursanta

2.1.1. Zarządzanie scenariuszami i modułami

Aplikacja trenera posiada możliwość kontroli aktualnie uruchomionych modułów, scenariuszy oraz poszczególnych kroków. Należy wziąć pod uwagę fakt, że poprawność działania wielu kroków jest uzależniona od wykonania poprzednich kroków, tworzących w aplikacji specyficzną sytuację potrzebną do poprawnego działania danego kroku. Z kolei scenariusze nigdy nie są wzajemnie zależne stąd ich uruchamianie w dowolnej kolejności jest bezpieczne dla aplikacji. Gdy kursant zakończy moduł trener może uruchomić kolejny moduł korzystając z listy modułów i scenariuszy przestanej z aplikacji kursanta do aplikacji trenera. Trener może także uruchomić dowolny scenariusz dla kursanta w dowolnym momencie.

2.1.2. Komunikacja z kursantem

Komunikacja z kursantem odbywa się za pomocą dedykowanego okna pozwalającego na wysyłanie wiadomości tekstowej. Wiadomość wyświetla się obok wirtualnego ekranu pilota, a kursant może ją zamknąć.

2.1.3. Podgląd widoku kursanta

W czasie trwania kursu trener ma cały czas podgląd tego, co widzi kursant. W razie gdyby użytkownik zatrzymał się na kroku scenariusza i nie mógł przejść dalej trener może to

zauważyć w swojej aplikacji. Istnieje też możliwość powiększenia widoku.

2.1.4. Wyświetlanie testowych informacji

W razie gdyby aplikacja kursanta nie działała poprawnie mamy możliwość wyświetlenia danych testowych aplikacji kursanta. Dane te zawierają takie informacje jak numer pilota, adres IP komputera czy liczbę przestanych ramek.

2.1.5. Zarządzanie niektórymi ustawieniami aplikacji kursanta

Kontrolki dostępne w aplikacji trenera pozwalają regulować poziom głośności lektora w aplikacji kursanta oraz zarządzać widocznością informacji pomocniczych (hint-ów). Te ostatnie zmieniane są grupowo tzn. trener może włączyć lub wyłączyć całą grupę hintów, ale nie pojedyncze. Dodatkowo, trener może zmieniać globalnie ustawienie języka aplikacji wszystkich kursantów, a także określić konkretne ID pilota, z którym aplikacja danego kursanta ma się połączyć.

2.2. Aplikacja kursanta

Aplikacja kursanta jest zbudowana modułowo, aby mieć możliwość uruchomienia dowolnego scenariusza w dowolnym momencie.

2.2.1. Budowa modułowa

Każda część kursu jest określona w aplikacji jako moduł. W moduły dzielą się dalej na scenariusze, które określają zbiór akcji, które razem tworzą pewien logiczny ciąg. Każdy scenariusz składa się z kroków które wykonuje kursant. Każdy moduł jest tworzony na osobnej scenie w Unity, a następnie jest ładowany w czasie działania aplikacji poprzez odpowiednie opcje w aplikacji trenera lub poprzez ukończenie poprzedniego modułu. Wyjątkiem jest moduł 6 (egzamin), który może być uruchomiony tylko poprzez aplikację trenera.

2.2.2. Systemy interakcji

Interakcja użytkownika z aplikacją odbywa się na dwa sposoby: akcje wywoływane kontrolerami i akcje wywoływane wzrokiem. Akcje wywoływane kontrolerem są związane z interakcją z otoczeniem, włączając w to używanie przycisków, przenoszenie rzeczy lub przewijanie listy błędów. Akcje wywoływane wzrokiem opierają się na stworzeniu możliwości wykrycia kierunku wzroku użytkownika i wykonanie określonych akcji, gdy użytkownik zauważył daną rzecz.

2.2.3. Implementacja interakcji z pilotem

Interakcja z aplikacją może odbywać się także za pomocą pilota. Akcje wywoływane są za pomocą fizycznych przycisków znajdujących się na pilocie, który przesyła informacje o stanie przycisków do aplikacji po protokole TCP. Wiadomości te są następnie przetwarzane w

aplikacji i odpowiednie akcje są wywoływane.

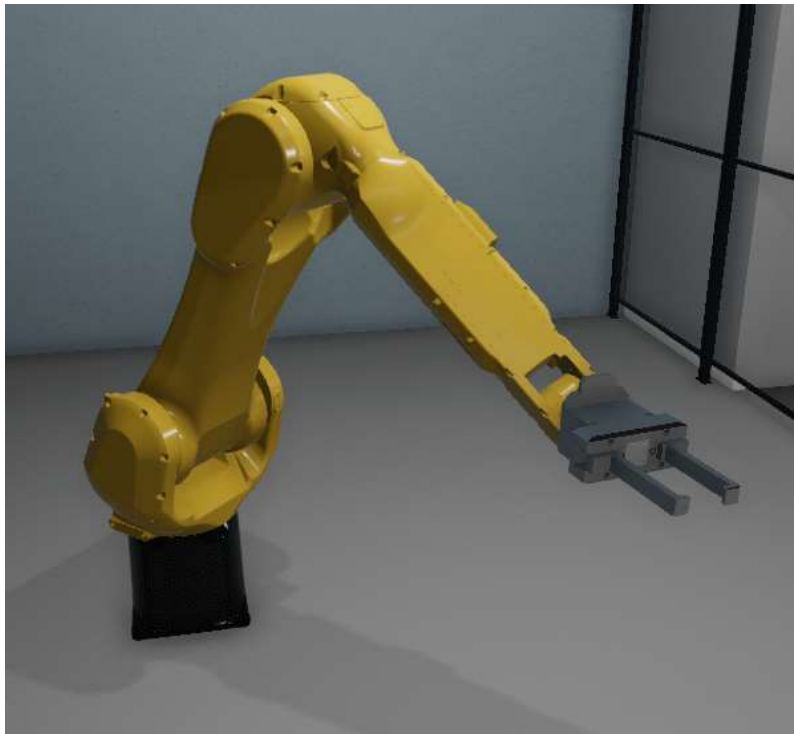
2.2.4. Wizualizacja obiektów

Systemy wizualizacji akcji na pilocie pozwalają na wizualizację rąk i palców na wirtualnym pilocie. Dzięki otrzymywanym danym jesteśmy w stanie stwierdzić z dużą dokładnością, gdzie palec kursanta dotyka ekranu pilota. Te informacje są następnie przetwarzane, a wirtualny model ręki dostosowywany do aktualnej realnej pozycji ręki. Warunkiem poprawnego działania tego systemu jest to, że użytkownik dotyka klawiatury pilota tylko dwoma palcami – jednym w lewej części klawiatury, drugim w prawej. Jest to naturalny sposób używania pilota, dlatego spełnienie tego warunku nie jest problematyczne, ale jeśli więcej palców dotknie jednej strony klawiatury, ich pozycja zostanie uśredniona. Ruch pilota w przestrzeni rzeczywistej jest przekazywany za pomocą trackera podłączonego do pilota. Tracker przekazuje informację o pozycji realnego pilota do świata wirtualnego, co pozwala odpowiednio dostosować pozycję modelu pilota w przestrzeni wirtualnej.

2.2.5. Elementy środowiska

Przestrzeń aplikacji jest podzielona na dwie części, między którymi kursant może się teleportować – wewnątrz celi, oraz przestrzeń zewnętrzną. Części te zazwyczaj oddzielone są metalową siatką, ale w celu zachowanie czytelności aplikacji, zastąpiono takie rozwiązanie przezroczystym panelem. Wewnątrz celi znajduje się wyłącznie sam robot oraz dwa podajniki, aktywujące się tylko w niektórych scenariuszach. W zewnętrznej części znajduje się szafka robota, kasetka, konsola z listą błędów, i kłódka oraz czapka bezpieczeństwa. W celi znajdują się również elementy które mogą przenosić się wraz z teleportującym się kursantem. Są to: kokpit zawierający przycisk teleportacji oraz przycisk przejścia do następnego kroku, oraz tablica zawierająca cele obecnego kroku oraz wideo pomocnicze.

W aplikacji używamy robota wykorzystującego system kości w ruchu liniowym oraz prosty system hierarchiczny w ruchu przegubowym.. Model wizualny robota, bazujący na robotach firmy FANUC, został wyeksportowany z programu Autodesk Inventor, a następnie dostosowany do potrzeb aplikacji przez grafika.



Rys 1 Model robota

2.3. Opis wybranych elementów implementacji systemu IT

2.3.1. Komunikacja aplikacji z pilotem

Za komunikację z pilotem odpowiada system połączenia na WiFi korzystający z przesyłania pakietów TCP/UDP i systemu multicast. Aplikacja po uruchomieniu nasłuchuje pakietów na sprecyzowanym IP, które jest jednocześnie w puli IP, która może przyjmować wiele pakietów na raz. Na te IP piloty wysyłają wiadomość ze swoim numerem identyfikacyjnym, który jednocześnie oznacza końcówkę ich numeru IP w sieci.

Gdy aplikacja otrzyma taki pakiet, wysyła wiadomość powitania do pilota, a następnie otwiera klienta TCP, który ma za zadanie odbierać wiadomości z pilota. Pilot łączy się z otwartym klientem i następuje wymiana wiadomości w ciągu trwania połączenia. Klient TCP jest uruchomiony na osobnym wątku w taki sposób, aby nie blokować wykonywania aplikacji.

W momencie utraty połączenia aplikacja będzie próbowała się połączyć z ostatnim pilotem, który był podłączony. Zapobiega to przypadkowym połączeniom, gdy zostanie uruchomiony więcej niż jeden pilot.

2.3.2. System kierowania wzroku kursanta

W czasie trwania kursu musimy sprawdzać, czy kursant spojrzął się na określone elementy otoczenia. W tym celu został stworzony system, który potrafi wykryć kierunek patrzenia. System składa się z dwóch elementów: tzw. Gazera i obiektów, które wchodzi z nim w interakcję.

Gazer jest dołączony do wirtualnego obiektu kamery, który porusza się razem z realnymi okularami VR. Dzięki możliwości przeniesienia ruchu i rotacji gogli do świata wirtualnego

Gazer za pomocą strzelania promienia w linii prostej od środka wirtualnej kamery jest w stanie wykryć, na jakie obiekt w świecie wirtualnym patrzy użytkownik.

Drugim elementem jest system obiektów, które wchodzi w interakcję z Gazerem. Każdy z tych obiektów ma podłączony komponent, który wykonuje sprecyzowane, generyczne akcje w momencie, gdy rozpoczniemy się na niego patrzeć, w momencie, którym przestaniemy na niego patrzeć, a także gdy patrzeliśmy się na niego przez określony czas. Dzięki temu jesteśmy w stanie wykonać dowolne akcje w każdym momencie spojrzenia użytkownika na dany obiekt.

2.3.3. Aktualizacja danych w aplikacji trenera

Gdy aplikacja trenera jest włączona informacje z aplikacji kursanta są wyświetlane w odpowiednich miejscach na interfejsie aplikacji trenera. W momencie połączenia aplikacji kursanta utworzony zostaje „kafelek” z widokiem z kamery użytkownika. Zaraz obok niego wyświetlone zostają najważniejsze informacje dotyczące statusu treningu – aktualny moduł, scenariusz i krok. Wszystkie informacje są aktualizowane w czasie rzeczywistym za pomocą komunikacji po protokole UDP na osobnym wątku. Dodatkowym elementem są dane testowe/typu debug, które pozwalają na diagnostykę ewentualnych problemów z połączeniem pilota do aplikacji kursanta. Dane te, wraz z dodatkowymi kontrolkami pozwalającymi zmieniać ustawienia aplikacji kursanta, dostępne są po rozszerzenia kafelka do widoku pełnoekranowego.

2.3.4. Warunki kroków i system celów

Aby określić czy kursant wykonał polecenia wymagane do zaliczenia kroku został przygotowany specjalny system warunków kroków, który umożliwia przejście do następnego kroku tylko jeśli wszystkie warunki obecnego kroku zostaną spełnione. Warunki kroku muszą umieszczone być w hierarchii sceny jako dzieci obiektu kroku do którego należą. Zostaną wtedy automatycznie zarejestrowane przez mechanizm kroków. Każdy skrypt warunku kroku posiada specjalne metody pozwalające implementować specyficzne zachowania przy inicjalizacji, zamknięciu kroku oraz w każdej klatce jego trwania. Posiada również zdarzenia pozwalające z poziomu inspektora dodawać metody wywoływane przy inicjalizacji i zamknięciu kroku, jak również za każdym razem gdy warunek kroku osiągnie lub straci status zaliczenia. Dzięki temu stworzony został bardzo elastyczny system pozwalający dowolnie łączyć warunki kroków w dowolne ciągi przyczynowo skutkowe.

Aby lepiej przekazać kursantowi informację o tym co należy zrobić w ramach kroku, stan wybranych warunków może być wyświetlony na specjalnej tablicy informacyjnej w postaci tekstowych komend. Komendy importowane są z zewnętrznego pliku i w zależności od tego czy odpowiadający im warunek został spełniony czy nie jest im nadawany odpowiedni kolor.



Rys 2 "Kafelek" z widokiem kursanta i informacjami nt. stanu kursu

2.3.5. Dynamiczne ładowanie tekstów i zdjęć/wideo zadań

Ze względu na możliwość zmian w tekstach i wideo/obrazach dotyczących kursów stworzony został system, który potrafi dynamicznie ładować pliki z określonych folderów umieszczonych w folderze danych aplikacji. Gdy zajdzie potrzeba zmiany tekstów nie potrzebna jest ponowna kompilacja aplikacji, wystarczy zmienić odpowiednie części pliku odpowiedzialnego za wyświetlanie tekstu i aplikacja sama wyświetli zaktualizowane dane.

System ładowania plików wideo/obrazów jest o party o foldery które odpowiadają kolejnym modułom/scenariuszom/krokami i plikach, które się w nich znajdują. Dzięki możliwości przeszukiwania folderów nie ma znaczenia czy umieścimy w folderze pliki obrazów czy wideo, system będzie w stanie załadować poprawne elementy do treningu.

3. Załączniki

Poniżej znajduje się lista załączników znajdujących się w folderze *Załączniki/*

- [Dokumentacja kodu aplikacji Kursanta/](#) - zawiera dokumentację kodu aplikacji kursanta wygenerowaną w formacie latex i html za pomocą narzędzia Doxygen.
- [Dokumentacja kodu aplikacji trenera/](#) - zawiera dokumentację kodu aplikacji trenera wygenerowaną w formacie latex i html za pomocą narzędzia Doxygen.
- [Ekran pilota](#) - Opis elementów występujących na wirtualnym ekranie pilota.
- [Hierarchia menu pilota](#) - Lista przedstawiająca hierarchiczną strukturę menu pilota.
- [Opis funkcjonalności przycisków pilota](#) - Opis funkcjonalności, jakie są przypisane do

poszczególnych przycisków klawiatury pilota w aplikacji.

- [Semantyka programów robota](#) – Opis składni instrukcji robota oraz przedstawienie niektórych procesów modyfikacji programu.