POLITECHNIKA POZNAŃSKA

WYDZIAŁ INFORMATYKI

MAGISTERSKA PRACA DYPLOMOWA

**APLIKACJA KOMPUTEROWA DO BEZDOTYKOWEJ OBSŁUGI DOKUMENTACJI MEDYCZNEJ Z WYKORZYSTANIEM PLATFORMY MICROSOFT KINECT**

**inż. Alicja Szulta**

### 

**Promotor:**

**dr inż. Janusz Pochmara**

Poznań, 2013

Spis treści

[1. Wstęp 3](#_Toc396196686)

[2. Platforma Microsoft Kinect 6](#_Toc396196687)

[2.1. Sprzęt 6](#_Toc396196688)

[2.2. Software Development Kit 10](#_Toc396196689)

[2.2.1. Naturalny Interfejs Użytkownika (NUI) 12](#_Toc396196690)

[2.2.2. Przechwytywanie strumieni 12](#_Toc396196691)

[2.2.3. Color Stream 13](#_Toc396196692)

[2.2.4. Depth Stream 15](#_Toc396196693)

[2.2.5. Skeleton Stream 16](#_Toc396196694)

[2.2.6. Interaction Stream 19](#_Toc396196695)

[2.3. Windows Developer Toolkit 21](#_Toc396196696)

[3. Windows Presentation Foundation (WPF) 23](#_Toc396196697)

[4. Platforma .NET i język C# 23](#_Toc396196698)

[5. Implementacja systemu 23](#_Toc396196699)

[5.1. Wybór zastosowanych technologii 23](#_Toc396196700)

[5.2. Funkcjonalności 24](#_Toc396196701)

[6. Spis ilustracji 25](#_Toc396196702)

[6.1. Schematy 25](#_Toc396196703)

[6.2. Rysunki 25](#_Toc396196704)

[7. Bibliografia 26](#_Toc396196705)

# Wstęp

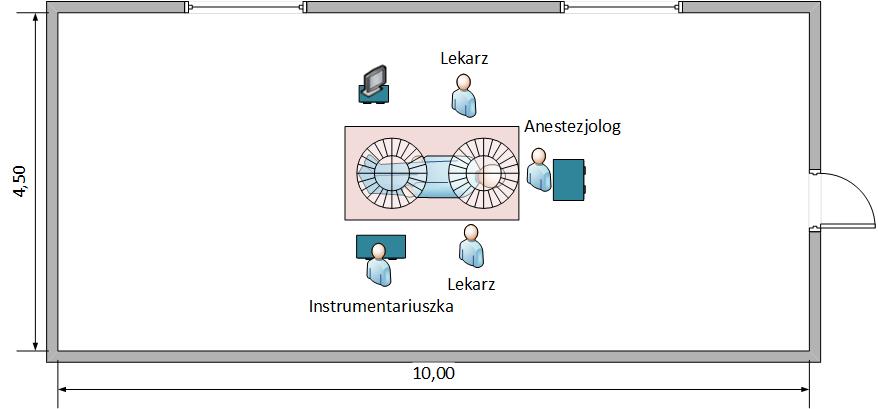
Wraz z rozwojem nauki i technologii, dostępne dla zwykłych użytkowników stają się coraz bardziej skomplikowane urządzenia, których obsługa staje się coraz bardziej intuicyjna, przyjazna i łatwa do nauki. Zmianie ulegają metody interakcji z urządzeniami - sterowanie dotykiem wypiera dotychczasowe klawiatury, sterowanie głosem staje się coraz powszechniej stosowane, komunikacja bezdotykowa (oparta na ruchach, gestach, mimice) już nikogo nie dziwi.

Ze względu na wymaganą sterylność, technologie bezdotykowe wkraczają w przestrzeń medyczną. Urządzeniem, które pozwala na bezdotykowe sterowanie aplikacjami jest Microsoft Kinect. Wśród projektów wykorzystujących Kinect'a w medycynie znajdują się: zastosowanie platformy Kinect do systemu rehabilitacji pozawałowej , Reflexion Rehabilitation Measurement Tool - system rehabilitacji po operacjach mięśni szkieletowych oraz systemu do przetwarzania dokumentacji medycznej podczas trwania operacji .

W ramach pracy magisterskiej zrealizowano projekt mający na celu usprawnienie korzystania z dokumentacji medycznej podczas zabiegów dokonywanych na sali operacyjnej jednego z poznańskich szpitali. W toku prac wstępnych nad projektem, odbyto konsultacje z lekarzami oraz zapoznano się z rzeczywistymi warunkami panującymi na sali operacyjnej.

Obecnie, podczas wykonywania procedur medycznych, dokumentacja medyczna dostępna jest w formie papierowej. W związku z wymaganiem sterylności podczas operacji, konieczna jest obecność na sali osoby, która może prezentować lekarzom dokumentację (wyniki badań, zdjęcia, prześwietlenia). W ramach projektu postanowiono zrealizować system, który pozwoli na wyświetlanie i prostą obróbkę (przybliżanie, oddalanie, obrót) dokumentacji medycznej w postaci elektronicznej. Co istotne, system musiał spełniać kryterium bezdotykowego sterowania. W związku z tymi wymaganiami zdecydowano się na wykorzystanie urządzenia MIcrosoft Kinect do stworzenia aplikacji komputerowej, która pozwoli na swobodne przeglądanie dokumentacji podczas zabiegów.

Poniższy schemat przedstawia rzut poziomy sali operacyjnej.



Schemat 1.1 Sala operacyjna - widok z góry (opracowanie własne)

Na środku sali znajduje się stół operacyjny, na którym leży operowany pacjent. Zlokalizowane bezpośrednio nad stołem lampy są źródłami bardzo intensywnego światła. Przy zabiegu/operacji może brać udział do dwóch lekarzy, anestezjolog oraz instrumentariuszka. Lekarze stoją po przeciwnych stronach stołu operacyjnego, anestezjolog znajduje się przy końcu stołu od strony głowy pacjenta, a instrumentariuszka obok jednego z lekarzy, od strony nóg pacjenta. Instrumentariuszka ma przed sobą stolik z narzędziami potrzebnymi przy wykonywanych procedurach medycznych (skalpele, nici etc.), natomiast anestezjolog przez cały czas operacji monitoruje stan pacjenta za pomocą specjalistycznej aparatury (na schemacie zaznaczona prostokątem za anestezjologiem). Ponadto, na sali znajduje się sprzęt medyczny, z którego korzysta się podczas zabiegu (na schemacie zaznaczone jako ekran po lewej stronie lekarza powyżej stołu operacyjnego). Warto zauważyć, że pacjent podłączony jest do wielu urządzeń medycznych, dlatego między stołem a aparaturą anestezjologa oraz pozostałą aparaturą przebiegają rozmaite połączenia, rurki, ssaki itp.

Na schemacie uwzględniono wymiary sali. Są to wymiary orientacyjne otrzymane od obsługi szpitala. Są one istotne ze względu na zasięg prawidłowej pracy urządzenia Kinect. Przedstawiono także umieszczenie okien. Okna są koloru mlecznego (nieprzezroczyste), przez co zapobiegają wpadaniu ostrego światła do sali operacyjnej. Ten fakt również jest bardzo ważnym czynnikiem, który należy wziąć pod uwagę rozważając umieszczenie urządzenia Kinect tak, żeby było to funkcjonalne dla obsługi medycznej oraz zapewniało prawidłowe działanie projektowanego systemu.

Podczas wykonywania operacji lekarze muszą zachować sterylność. Oznacza to, że nie mają możliwości sterowania żadnymi urządzeniami przy użyciu rąk, ponieważ są one w bezpośrednim kontakcie z organizmem pacjenta. Wśród obecnych na sali osób jest tylko jedna, która może dotykać różnych sprzętów, ale ma ona obowiązki podczas wykonywania procedur medycznych, stąd nie można zaangażować jej do sterowania systemem komputerowym do wyświetlania dokumentacji medycznej. Jest to przyczyna, dla której używanie myszki czy klawiatury - standardowych urządzeń do sterowania komputerem - nie jest możliwe podczas zabiegu. Jednakże, podczas przygotowania do wykonywania zabiegu odbywa się inicjalizacja sprzętu medycznego, kiedy to kontakt bezpośredni jest możliwy. Dokumentacja medyczna znajduje się na elektronicznych nośnikach danych (płyty CD).

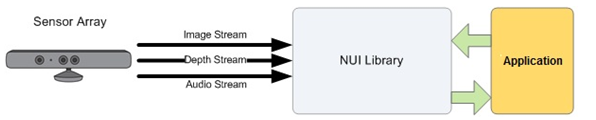
Odbywając konsultacje z lekarzami zapytano o ich potrzeby względem projektowanego systemu. Na podstawie zebranych informacji sformułowano następujące wymagania projektowe:

* system musi być sterowany bez bezpośredniego kontaktu (bezdotykowo),
* podczas inicjalizacji systemu przed zabiegiem dopuszczalny jest bezpośredni kontakt ze sprzętem (a więc sterowanie myszką, klawiaturą),
* system ma zapewniać funkcjonalność powiększania, pomniejszania dokumentacji medycznej (zdjęć) oraz ich obracanie,
* ustawienie urządzenia Microsoft Kinect na sali operacyjnej musi zapewniać funkcjonalność systemu dla użytkowników (nie może kolidować z aparaturą medyczną i wymagać zbytniego przemieszczania się po sali),
* ustawienie urządzenia Microsoft Kinect na sali operacyjnej musi zapewniać prawidłowe działanie systemu (problem wielu źródeł intensywnego światła oraz wielu osób przebywających na sali i biorących aktywny udział w przeprowadzanej operacji).

Sposoby rozwiązania oraz argumentacja przyjętych metod realizacji powyższych wymagań opisana jest w zasadniczej części niniejszej pracy.

# Platforma Microsoft Kinect

Urządzenie Kinect jest platformą wykrywania ruchu. Na stronie internetowej producenta można przeczytać, że Kinect for Windows "jest zestawem technologii, które umożliwiają ludziom naturalną komunikację z komputerami; czujnik oraz darmowy zestaw bibliotek i narzędzi programistycznych (SDK) Kinect for Windows dostarczają deweloperom fundament wymagany do tworzenia i rozwijania interaktywnych aplikacji, które reagują na ludzki naturalny ruch, gesty i polecenia głosowe" (tłumaczenie własne). Można zatem stwierdzić, że zestaw Kinect składa się z dwóch zasadniczych elementów - platformy sprzętowej oraz platformy programistycznej. Dane, otrzymane w wyniku przetwarzania otoczenia przez czujniki platformy Kinect, przekazywane są do aplikacji. Dzięki wykorzystaniu naturalnego interfejsu użytkownika (NUI - ang. *Natural User Interface*), sygnały wejściowe urządzenia Kinect oraz komendy sterujące programem są przyjazne i intuicyjne. Te zasady współdziałania platformy programistycznej i sprzętu przedstawia poniższy schemat.



Schemat . Interakcja sprzętu i oprogramowania z aplikacją (Microsoft Corporation, 2014)

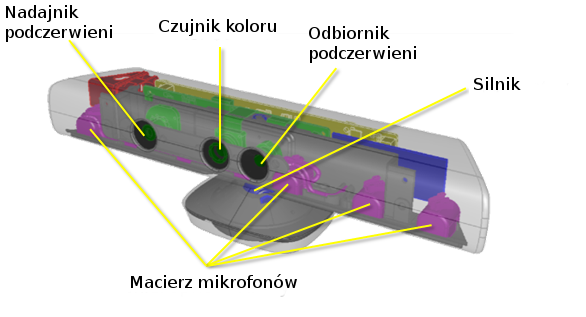
Ponadto, przydatnym narzędziem w pracy nad programami wykorzystującymi urządzenie Kinect for Windows jest Windows Developer Toolkit. Wszystkie te komponenty omówione są w podrozdziałach niniejszego rozdziału.

## Sprzęt

Urządzenie Kinect składa się z następujących podzespołów:

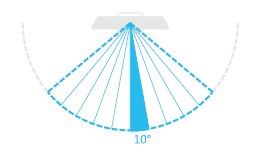
* macierz mikrofonów kierunkowych,
* promiennik podczerwieni,
* odbiornik podczerwieni,
* kamera RGB,
* akcelerometr,
* silnik

Ich rozmieszczenie obrazuje poniższy schemat:



Schemat 2.2 Podzespoły urządzenia Kinect for Windows - opracowanie własne na podstawie (Microsoft Corporation, 2014)

Macierz mikrofonów składa się z czterech mikrofonów, posiada 24-bitowy przetwornik analogowo-cyfrowy. Układ wykrywa dźwięki z obszaru w zakresie pięćdziesięciu stopni na prawo i pięćdziesięciu stopni na lewo od środka czujnika. Obszar ten można podzielić na dziesięć dziesięciostopniowych fragmentów, co pozwala określić które dźwięki pochodzą z precyzyjnie określonego kierunku. Jest to rozwiązanie szczególnie przydatne do wybierania przetwarzanych dźwięków na podstawie położenia ich źródła.

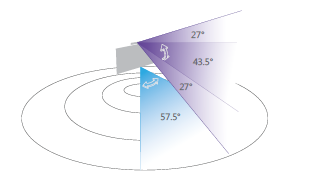


Schemat 2.3 Zakres wykrywania dźwięków (Microsoft Corporation, 2013)

Ponadto, dopuszczalne jest wprowadzenie progu natężenia dźwięku - Kinect umożliwia odfiltrowanie dźwięków poniżej 20 decybeli. Co więcej, dźwięki pochodzące zza czujnika są tłumione o 6dB. Zapewniona jest obsługa dźwięku w formacie PCM z częstotliwością próbkowania 16 kHz, z rozdzielczością 24 bitów na próbkę reprezentującym dźwięk monofoniczny (1 kanał). Z poziomu programowego dostępne są funkcje śledzenia głosu jednego ze śledzonych graczy oraz śledzenia najgłośniejszego źródła dźwięku.

Promiennik i odbiornik podczerwieni razem stanowią podsystem sensora głębokości (ang. *depth camera*), który pozwala na wykrywanie odległości pomiędzy przedmiotami i osobami znajdującymi się w zasięgu ich działania a Kinect'em. Zmierzona wartość wyrażona jest w milimetrach. Układ sensora głębokości przekazuje obraz z prędkością 30 FPS w jednej z dostępnych rozdzielczości - 320x420, 640x480 oraz 80x60. Zasięg pracy czujnika stanowi przestrzeń zakreślona między 43,5 stopniami w pionie (+/- 27 stopni względem poziomu czujnika) i 57,5 stopniami w poziomie.

Kamera RGB dostarcza obraz z taką samą prędkością jak sensor głębokości (30 FPS). Zakres pracy czujnika jest taki sam jak dla sensora głębokości. Schematycznie obrazuje to poniższy rysunek.



Schemat . Zasięg kątowy pracy czujnika głębokości i RGB

W związku z fizycznymi ograniczeniami czujników obrazu oraz głębokości, efektywne korzystanie z Kinect'a wymaga przestrzegania pewnych wymagań dotyczących obszaru działania użytkownika. Dostępne są dwa tryby działania sensora głębokości - domyślny (ang. *default*) oraz bliski (ang. *near*). Tryb działania sensora wpływa na to jakie odległości użytkownika od urządzenia Kinect są dopuszczalne i prawidłowo rozpoznawalne. W trybie domyślnym Kinect ma zakres pracy w odległości od 0.8 do 4 metrów, jednak dla najlepszej efektywności otrzymywanych sygnałów zalecane jest użytkowanie Kinect'a w odległości 1.2 do 3.5 metra od czujnika. Czujnik głębokości pracuje poprawnie także dla dystansów większych niż zalecane, ale informacje o śledzeniu i rozpoznawaniu graczy mogą być już niewiarygodne. W trybie pracy bliskiej (ang. *near mode*) zalecana odległość od czujnika zapewniająca stabilną pracę wynosi od 0.8 do 3 metrów, zaś fizycznie dopuszczalny obszar zamyka się między 0.4 a 3 metry od Kinect'a.

Podczas prac nad niniejszym projektem, gdy próbowano ustawić w programie tryb *near mode*, okazało się, że model urządzenia Kinect, którym dysponowała autorka nie jest egzemplarzem dedykowanym do użytku z systemem Windows (a co za tym idzie z pakietem bibliotek Kinect SDK) ale do użytku z konsolą Xbox. W związku z tym, tryb pracy bliskiej jest dla niego niedostępny. Poniżej przytoczono komunikat, który pojawił się podczas debuggowania błędu:

The Kinect sensor plugged into your computer is for use on the Xbox 360. You may continue using your Xbox 360 Kinect sensor on your computer for development purposes. Microsoft does not guarantee full compatibility for Kinect for Windows applications and the Xbox 360 Kinect Sensor.

Powyższa informacja oznacza, że urządzenie Kinect podłączone do komputera jest dedykowane do użytku z konsolą Xbox 360; dopuszczalne jest używanie go w celu rozwoju oprogramowania, jednak Microsoft nie gwarantuje pełnej kompatybilności aplikacji dedykowanych dla Kinect for Windows z czujnikiem Kinect dedykowanym dla Xbox 360 (tłumaczenie własne). W związku z tym, dla pełnej sprawności programu, zaleca się korzystanie z czujnika Kinect for Windows.

Kinect wyposażony jest także w trójosiowy akcelerometr, który może pracować w trybach 2G, 4G oraz 8G (domyślnie skonfigurowany na 2G). Obecność akcelerometru na platformie Kinect pozwala na ustalenie aktualnej orientacji układu względem kierunku grawitacji. Zaimplementowane w SDK metody pozwalają ponadto ustalić orientację względem poziomu. Daje to możliwość manipulacji kątem pochylenia zestawu Kinect for Windows za pomocą silniczka, który wmontowany jest w podstawkę urządzenia. Układ pozycjonowania Kinect'a działa z dokładnością jednego stopnia, jednak jak przestrzega producent, akcelerometr jest czuły na temperaturę pracy. W związku z tym, przekroczenie normalnej temperatury pracy (5-35o Celsjusza ) może skutkować w maksymalnie trzystopniowym odchyleniu od deklarowanej dokładności pomiaru akcelerometru (1o). Na szczęście, dla konkretnego urządzenia odchylenie to ma zawsze stałą wartość.

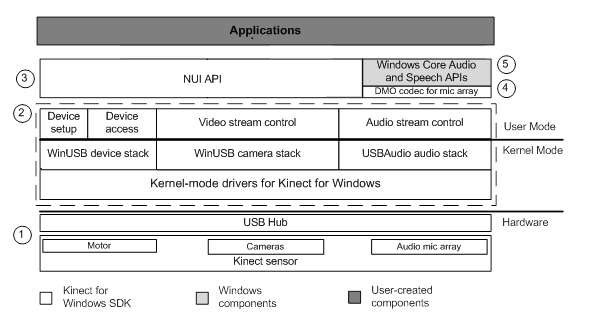
Podczas korzystania z Kinect'a w aplikacjach systemu Windows trzeba pamiętać, że ponieważ czujnik jest wyposażony jedynie w procesor sygnałowy (ang. *Digital Signal Processor*) poddający obróbce sygnał dźwiękowy z mikrofonów, przetwarzanie danych odbywa się po stronie sterownika Kinect'a na komputerze.

## Software Development Kit

Kinect for Windows Software Development Kit to zestaw pomocy do obsługi programowej czujnika na komputerach z systemem operacyjnym Windows. Zestaw ten zawiera:

* sterowniki i dokumentację techniczną,
* dokumentację i zestaw źródłowych API w języku C++, C# oraz VisualBasic,
* przykładowe projekty przedstawiające wykorzystanie dostarczonych API.

Strukturę dostarczonego SDK przedstawia niniejszy schemat:



Schemat 2.5 Architektura SDK (Microsoft Corporation, 2014)

Bloki z białym tłem oznaczają komponenty Kinect for Windows SDK, bloki z jasnoszarym tłem symbolizują komponenty systemu Windows, a bloki z ciemnoszarym tłem obrazują komponenty użytkownika (tutaj - aplikację wykorzystującą urządzenie Kinect).

Obszar oznaczony numerem jeden odnosi się do sprzętowej budowy platformy Kinect omówionej szerzej w poprzednim podrozdziale. Moduły obszaru numer dwa współpracują ze sobą w ramach sterowników urządzenia Kinect. Dzięki driverom macierz mikrofonów może być używana przez użytkownika systemu /Windows jako standardowe wejście audio. Obszar trzeci to obszar NUI (ang. *Natural User Interface*), na który składają się komponenty audio i wideo Kinect'a. Blok czwarty składa się z kodeku DMO[[1]](#footnote-1) (ang. *DirectX Media Object*) dla macierzy mikrofonów. Blok piąty to blok standardowych API Windows 7 dostępnych także dla aplikacji desktopowych Windows 8. Z powyższego schematu wynika, że czujnik Kinect wraz z dostarczonymi bibliotekami SDK tworzy całościową platformę programistyczną zintegrowaną na poziomie sprzętowym.

W ramach pracy magisterskiej korzystano z Kinect for Windows SDK w wersji 1.8. Dla tej wersji zestawu producent deklaruje wsparcie dla następujących systemów operacyjnych:

* Windows 7
* Windows 8
* Windows Embedded Standard 7
* Windows Embedded Standard 8

W przypadku gdy posiadany przez użytkownika system Windows 7 występuje w wersji Windows 7N albo Windows 7 KN, wymagane jest doinstalowanie zestawu Media Feature Pack . Ponadto, występują też wymagania sprzętowe dla prawidłowego działania dostarczonych narzędzi:

* procesor 32-bit (x86) lub 64-bit (x64)
* procesor Dual-core, 2.66-GHz lub szybszy
* szyna USB 2.0 dedykowana dla urządzenia Kinect
* 2 GB pamięci RAM
* karta graficzna wspierająca DirectX 9.0c

Zestaw SDK dostępny jest na zasadach wolnej licencji dla użytkowników niekomercyjnych. Szczegółowe zasady określone są w dokumencie EULA dostępnym na stronie Microsoftu. Warto jednak wspomnieć, iż na mocy licencji dopuszczalne jest projektowanie, tworzenie i testowanie oprogramowania Kinect for Windows Application przy użyciu urządzenia Kinect for Xbox 360, jednakże użytkownik końcowy, aby legalnie korzystać ze stworzonego programu, musi posiadać sensor Kinect dedykowany dla systemu Windows (Microsft Corporation, brak daty). Jest to także związane z faktem, iż podczas tworzenia i testowania programu obsługującego urządzenie Kinect, na komputerze dewelopera czujnik Kinect for Xbox 360 jest warunkowo obsługiwany. Na innych maszynach, próba włączenia tego samego programu poprzez podłączenie Kinect'a dedykowanego do konsoli Xbox zostanie uniemożliwiona - urządzenie nie zostanie wykryte jako obsługiwane (Microsoft Corporation, 2014).

### Naturalny Interfejs Użytkownika (NUI)

Naturalny interfejs użytkownika jest to interfejs użytkownika, który zapewnia łatwe i szybkie nauczenie się obsługi dzięki temu, że nie jest konieczne używanie sztucznych metod i urządzeń do interakcji z aplikacją, ale wykorzystywane są naturalne metody komunikacji (np. mowa, gesty) . Programista korzystający z Kinect for Windows API buduje interfejs programu przy użyciu danych przechwytywanych przez NUI. Dane te zawierają się w strumieniu danych audio oraz w strumieniu obrazu i strumieniu głębokości. Te trzy typy sygnałów wejściowych dają możliwość stworzenia interfejsu programu przyjaznego użytkownikowi.

Na podstawie pomiarów sensora głębokości oprogramowanie obsługujące urządzenie Kinect rozpoznaje i śledzi ludzkie ciało. System jest w stanie rozpoznać nie więcej niż sześciu użytkowników jednocześnie, z czego dwóch może być aktywnie śledzonych (rozpoznawanie i śledzenie użytkowników jest szerzej omówione w kolejnych częściach pracy). Kompatybilność z Microsoft Speech API umożliwia implementację komend słownych, za pomocą których dostępne jest sterowanie aplikacją korzystającą z Kinect'a. Ponadto, dzięki dostępnym strumieniom danych zrealizowano śledzenie twarzy przy pomocy omawianego urządzenia. Wszystkie te rozwiązania dostępne są dla deweloperów w Kinect for Windows SDK.

W kolejnych sekcjach opisane są strumienie danych, z których korzystano na potrzeby pracy. Jako że nie używano przetwarzania dźwięków, nie omawiano strumienia audio. W celu głębszego poznania specyfiki tego strumienia odsyłam do dokumentacji dostępnej na stronie Microsoftu: http://msdn.microsoft.com/en-us/library/jj131026.aspx.

### Przechwytywanie strumieni

Strumienie danych dostępnych z czujników platformy Kinect można programowo przechwytywać dzięki API dostarczonemu wraz z Kinect for Windows SDK. Zestaw bibliotek daje możliwość uzyskania dostępu do danych w dwojaki sposób - poprzez zdarzenia wywoływane przy otrzymaniu każdej klatki z odpowiednimi danymi oraz poprzez mechanizm odpytywania o kolejne klatki.

Metoda odpytywania polega na tym, aby po inicjalizacji danego strumienia odpytywać go ręcznie o nową ramkę danych. Zapytanie trwa tak długo jak zdefiniowany okres czasu. Jeżeli okres ten minie, a nowa ramka danych nie nadejdzie, zwracana jest wartość *null* (w języku C#). Dzieje się to za pomocą metody *OpenNextFrame* wywołanej dla żądanego typu strumienia danych.

Aby przechwytywać ramki z danymi metodą zdarzeń (ang. *events*), należy zapisać się na wywoływane przez strumień zdarzenie. W języku C# dzieje się to przez uchwyt *ColorFrameReady, DepthFrameReady, SkeletonFrameReady* albo *AllFramesReady* obiektu klasy *KinectSensor*. Trzy pierwsze wymienione uchwyty pozwalają na przechwycenie klatek danego typu (odpowiednio - kolorowego/podczerwonego, głębokości, szkieletu). Uchwyt *AllFramesReady* zawiera wszystkie trzy typy klatek. Zdarzenia typu *ColorFrameReady*, *DepthFrameReady* oraz *SkeletonFrameReady* wywoływane są w momencie, gdy dostępna jest nowa klatka danego typu - wówczas następuje zapytanie o tę nową ramkę danych (*OpenNextFrame*). Natomiast zdarzenie *AllFramesReady* wywoływane jest, gdy dostępne są nowe klatki dla każdego typu strumienia . Zdarzeniowy mechanizm przechwytywania danych z czujników urządzenia Kinect jest charakterystyczny dla platformy Windows Presentation Foundation, której użyto do realizacji projektu.

Warto zauważyć, iż dla każdy strumień można obsługiwać za pomocą dowolnie wybranej metody. W związku z tym, dopuszczalny jest stan kiedy strumień obrazu kolorowego obsługiwany będzie mechanizmem zdarzeń, a strumień głębokości zapytaniami. Jednakże, w przypadku zapisania się na zdarzenie *AllFramesReady*, manualne odpytywanie któregokolwiek ze strumieni powoduje wystąpienie wyjątku typu *InvalidOperationException* .

### Color Stream

Strumień obrazu kolorowego (RGB) umożliwia przechwycenie zwykłego obrazu z kamery Kinect'a z prędkością od 12 do 30 klatek na sekundę. Obraz dostępny jest w jednym z trzech zdefiniowanych formatów - RGB, YUV oraz Bayer. Niezależnie od wybranego formatu obraz zawsze reprezentuje te same dane, gdyż przetwarzanie do wybranego formatu odbywa się z tych samych danych źródłowych pochodzących z kamery . Podczas inicjalizacji strumienia w kodzie programu należy wybrać rozdzielczość (dostępne są 80x60, 320x240, 640x480 oraz 1280x960 pikeli) i format przechwytywanego obrazu. Tak zdefiniowanych parametrów nie można zmieniać podczas przechwytywania danych - aby je przeformułować należy wyłączyć i ponownie włączyć strumień danych obrazu kolorowego (z nowymi wartościami parametrów).

Klatka obrazu pochodzącego z kamery RGB urządzenia Kinect składa się z pikseli. Ich ilość zależy od wybranej rozdzielczości, zatem układ współrzędnych (x,y) obrazu związany jest z jego rozmiarem. Każdy piksel posiada informację o wartościach kolorów czerwonego, zielonego i niebieskiego.



Rysunek 2.1 Obraz kolorowy (źródło własne)

Strumień RGB posiada także w sobie strumień pochodzący z czujnika podczerwieni. Podczas przechwytywania strumienia można korzystać tylko z jednego obrazu na raz - albo z czujnika RGB, albo z czujnika podczerwieni .

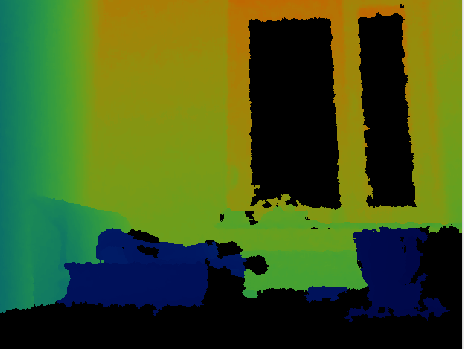


Rysunek 2.2 Obraz podczerwony (źródło własne)

### Depth Stream

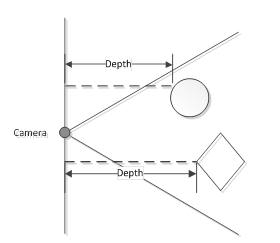
Strumień głębokości obrazu dostępny jest pod nazwą *DepthStream*. Każdy piksel ramki typu *DepthImage* zawiera informację o odległości wykrytego obiektu (najbliższego w danym punkcie obszaru widzenia sensora głębokości) od płaszczyzny czujnika wyrażoną w milimetrach. Ponadto, każdy piksel posiada informację o tym, który użytkownik na nim występuje. Jest to określone za pomocą indeksu w zakresie od 0 do 6, ponieważ Kinect zapewnia wykrywanie do sześciu użytkowników (ang. *player segmentation data*). Indeksowanie użytkowników dostępne jest tylko wtedy, kiedy oprócz strumienia głębokości zainicjalizowany jest strumień informacji o szkielecie (ang. *skeleton stream*) . Wartości indeksu w zakresie 1-6 odpowiadają indeksom użytkowników w tablicy danych szkieletu pomniejszonym o jeden, wartość 0 oznacza, że w danym miejscu nie ma żadnego z użytkowników Kinect'a.

Obraz przechwytywany ze strumienia głębokości może przyjąć jedną z dostępnych rozdzielczości - 640x480, 320x240 albo 80x60 pikseli. Zakres pracy bliskiej (ang. *near mode*) oraz domyślnej (ang. *default mode*) omówione są w sekcji dotyczącej sprzętowych uwarunkowań platformy Kinect.



Rysunek 2.3 Obraz głębokości (źródło własne)

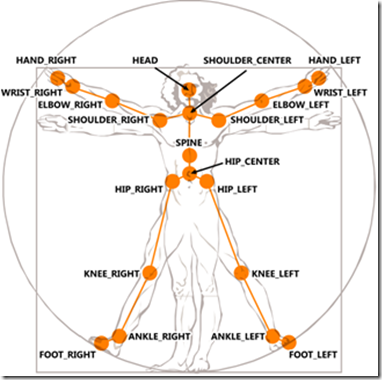
Dane otrzymywane ze strumienia głębokości zawierają odległość do najbliższego wykrytego obiektu w konkretnym punkcie. Zależność tę ilustruje schemat:



Schemat 2.6 Układ współrzędnych dla strumienia głębokości (Microsoft Kinect, 2014)

### Skeleton Stream

Dane strumienia szkieletu obliczane są na podstawie informacji ze strumienia RGB z czujnika podczerwieni. Do danych strumienia szkieletu należy położenie stawów, ich orientacja oraz rodzaj. Urządzenie umożliwia wykrywanie do sześciu osób (szkieletów), przy czym zapewnia śledzenie maksymalnie dwóch. Śledzenie oznacza wykrywanie i przetwarzanie informacji o wszystkich dostępnych ("widzialnych" dla czujnika) stawach użytkownika. Poniższy rysunek przedstawia wszystkie obsługiwane przez SDK stawy oraz ich umiejscowienie na ciele człowieka.



Rysunek 2.4 Układ stawów - Skeleton Stream (Raiten, 2014)

Praca strumienia szkieletu może odbywać się w jednym z dwóch dostępnych trybów - w trybie pełnego szkieletu (ang. *full skeleton mode*) oraz w trybie siedzącym (ang. *seated mode*). Śledzenie w trybie pełnego szkieletu rozpoznaje wszystkie 20 stawów, zaś w trybie siedzący skupia się tylko na dziesięciu górnych stawach.



Rysunek . Tryb siedzący

Interesującym zagadnieniem jest filtrowanie i wygładzanie położenia stawów przy użyciu parametrów typu *TransformSmoothParameters*. Konieczność skorzystania z funkcji wygładzania położenia stawów wynika z błędów pomiaru sensorów oraz występujących w otoczeniu szumów. Wymienione niedokładności są konsekwencją wielu czynników determinujących wiarygodność pomiarów - od dynamicznie zmieniającego się oświetlenia po kształty i rozmiary użytkownika . Inicjalizując w kodzie programu wykorzystującego platformę Kinect strumień szkieletu, można uwzględnić parametry filtrowania. Struktura typu *TransformSmoothParameters* składa się z następujących pól:

* *Smoothing* - parametr wygładzania, wzrost parametru wpływa na wzrost opóźnień, zakres wartości 0-1,
* *Correction* - korekcja, niższe wartości korygują wolniej, ale płynniej, wyższe wartości korygują szybko (i skokowo), zakres wartości 0-1,
* *Prediction* - predykcja, ilość klatek do przewidzenia z wyprzedzeniem, zakres wartości - od zera w górę,
* *JitterRadius* - kąt fluktuacji, kąt wyrażony w metrach do redukcji fluktuacji,
* *MaxDeviationRadius* - maksymalny kąt wyrażony w metrach, o który mogą się odchylać wartości przefiltrowane od wartości surowych (wejściowych) .

W ramach pracy wartości parametrów dobrano eksperymentalnie. Ostateczna postać omawianej struktury przyjęła następujące współczynniki:

Smoothing = 0.75f,

Correction = 0.07f,

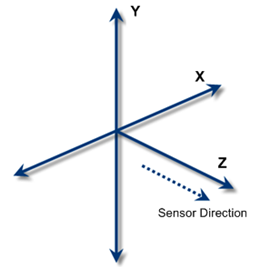
Prediction = 0.08f,

JitterRadius = 0.08f,

MaxDeviationRadius = 0.07f.

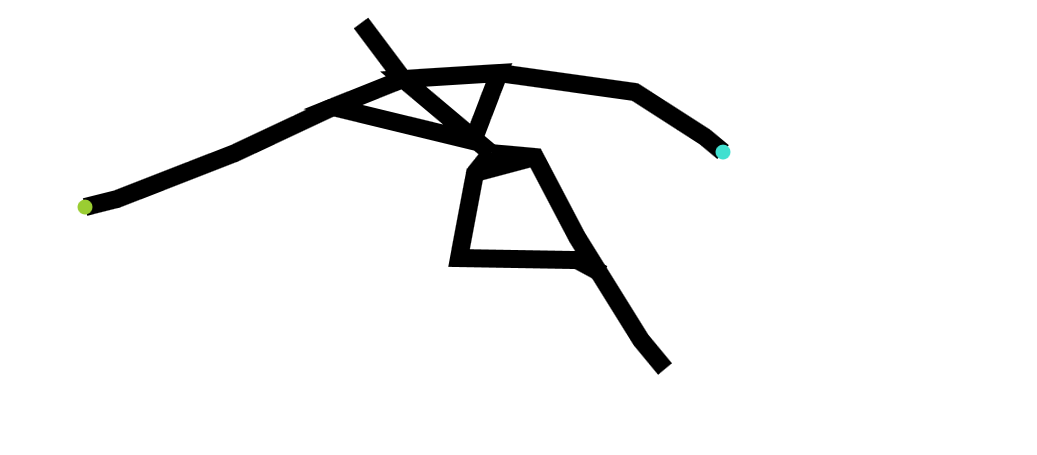
Każdy z rozpoznanych szkieletów posiada właściwość stanu śledzenia (ang. *tracking state*) określoną jednym z trzech stanów - śledzony (ang. *tracked*), tylko pozycja (ang. *position only*) albo nieśledzony (ang. *not tracked*). Śledzenie użytkownika (szkieletu) oznacza pozycjonowanie jego stawów w przestrzeni, tylko pozycja oddaje położenie gracza w przestrzeni, jednak bez szczegółowych danych na temat jego poszczególnych części ciała. Podobnie, każdy ze stawów śledzonych szkieletów posiada określony stan - śledzony, nieśledzony oraz implikowany (ang. *inferred*). Stan implikowany (wnioskowany, dedukowany) oznacza, że dany staw nie jest bezpośrednio "widoczny" dla sensora Kinect, dlatego jego pozycja jest przypuszczana na podstawie położenia i kątów skręcenia pozostałych stawów. Jest to stan obarczony dużą niepewnością, dlatego wszelkie przetwarzanie położeń i kątów między stawami powinno się odbywać dla stawów w stanie śledzenia.

Strumień danych szkieletu związany jest przestrzenią trójwymiarową. Każdy śledzony punkt typu szkieletowego (ang. *SkeletonPoint*) posiada pozycję wyrażoną w układzie (x,y,z). Jednostką miary w omawianym układzie jest metr (należy o tym pamiętać, ponieważ w układzie strumienia głębokości wartości wyrażone są w milimetrach). Orientację tego układu przedstawia schemat.



Schemat . Układ współrzędnych strumienia szkieletu

Jest to prawoskrętny układ współrzędnych kartezjańskich. Oś z skierowana jest w stronę, w którą skierowany jest czujnik. Stąd, współrzędna z określa odległość użytkowników od urządzenia Kinect we współrzędnych szkieletowych. Wartości na osi y rosną w górę względem położenia czujnika, natomiast wartości dla osi odciętych x wzrastają wraz z przesuwaniem się obiektów bardziej w lewą stronę Kinect'a. .

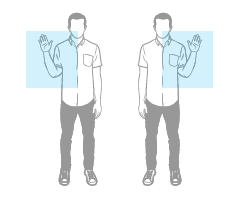


Rysunek . Śledzony szkielet (źródło własne)

### Interaction Stream

Kolejnym interesującym strumieniem, którym dysponuje urządzenie Kinect jest strumień interakcji (ang. *interaction stream*). Jest to o tyle ciekawy zestaw danych, że powstaje na podstawie danych ze strumienia szkieletu oraz strumienia głębokości przetworzonych przez złożone algorytmy, co pozwala uzyskać funkcję śledzenia dłoni i rozpoznawania gestów . Dostęp do strumienia interakcji w aplikacji można uzyskać, jeżeli użyje się dostarczonej z zestawem SDK biblioteki KinectInteraction. Ponadto, w bibliotece kontrolek obsługujących urządzenie Kinect (występuje w SDK) Microsoft.Kinect.Toolkit.Controls.dll występują komponenty wykorzystujące strumień interakcji do obsługi kursora dłoni. Do najważniejszych z nich należy zaliczyć wskaźnik dłoni (ang. *hand pointer*), który wykorzystywany jest jako odpowiednik kursora myszki w sterowaniu aplikacją windowsową wykorzystującą platformę Kinect. Bardzo korzystnym z deweloperskiego punktu widzenia jest fakt, iż biblioteki zawierające kontrolki korzystające ze strumienia interakcji dostępne są nie tylko z poziomu skompilowanej biblioteki (ang. *dynamic-link library*), ale dostarczone są także ich kody źródłowe. Dzięki temu, można dostosować kontrolki do własnych potrzeb - z tego rozwiązania skorzystano na potrzeby niniejszej pracy, co zostanie omówione w dalszych frgamentach.

Strumień interakcji generuje strefę fizycznej interakcji (ang. *Physical Interaction Zone, PhIZ*). Jest to dynamicznie obliczany obszar, osobny dla każdej z rozpoznanych dłoni, związany z położeniem ciała użytkownika. Położenie i rozmiar stref interakcji dla poszczególnych dłoni użytkownika przedstawia schemat.



Schemat 2.8 Fizyczna strefa interakcji (ang. *Physical Interaction Zone*) (Microsoft Corporation, brak daty)

Wraz z przemieszczaniem się użytkownika aplikacji, jego strefa interakcji przemieszcza się razem z nim. Dzięki temu uzyskana jest ergonomia ruchów - ruchy ręki wewnątrz PhIZ rzutowane są na całą kontrolkę, dzięki czemu, aby kursorem dłoni osiągnąć skrajną krawędź ekranu nie trzeba sięgać ręką daleko w bok.

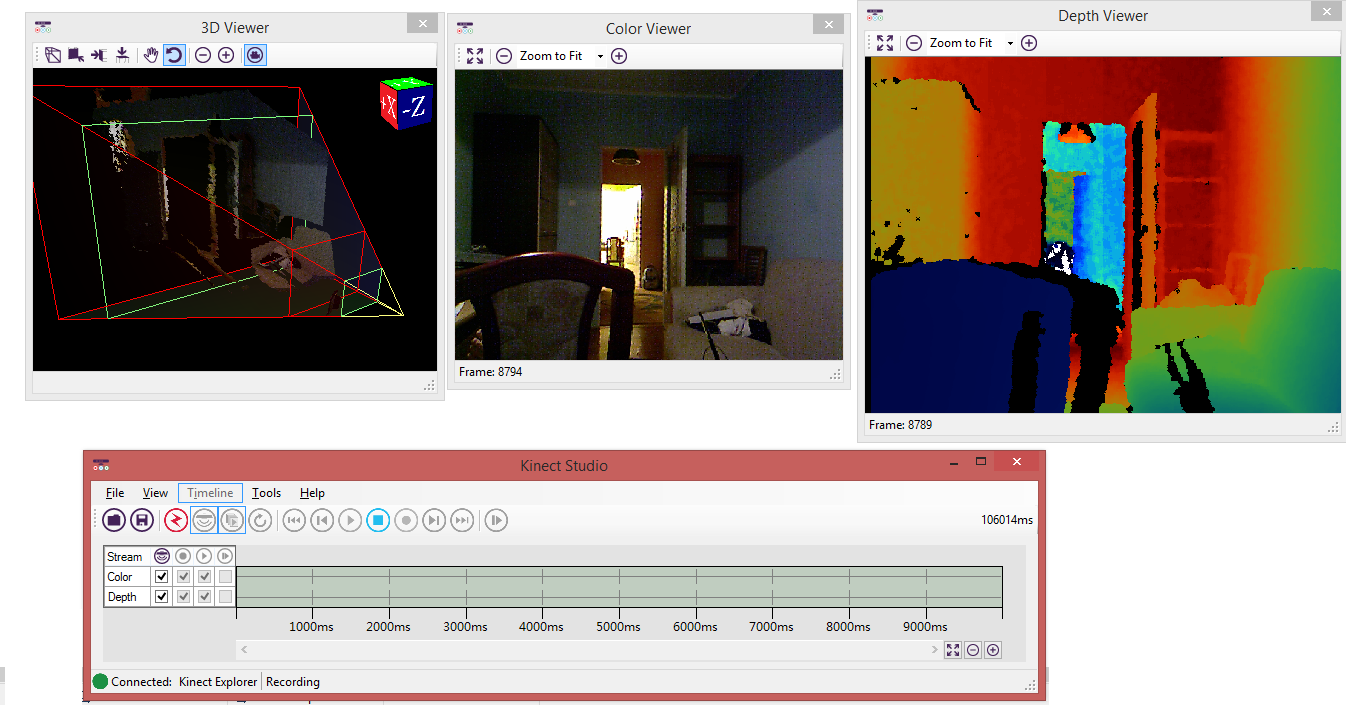
Poprzez zastosowane w strumieniu interakcji złożonych algorytmów, wybierana jest dłoń główna, za pomocą której odbywa się sterowanie aplikacją. Zasadniczo sprowadza się to do wyboru tej dłoni, która jako pierwsza została uniesiona. Podczas użytkowania aplikacji, dłoń główna (ang. *primary hand*) jest zmieniana dynamicznie - w zależności od ruchów i postaw przyjmowanych przez użytkownika programu. Domyślnie wspierane jest operowanie jedną dłonią na raz.

Strumień i bibliotek interakcji implementują także trzy rodzaje interakcji dłoni (kursora) z aplikacją. Są to - uchwycenie i puszczenie (ang. *grip and release*), wciśnięcie (ang. *press*) oraz przewijanie (ang. *scroll*). Dzięki tym narzędziom zaimplementowano naturalny interfejs użytkownika (NUI), który opiera się na komunikatach znanych z życia codziennego, przez co bardzo intuicyjnych dla użytkownika programu.

## Windows Developer Toolkit

Kinect for Windows Developer Toolkit (w ramach pracy używano w wersji 1.8.0) jest środowiskiem, w którym zebrano wszystkie komponenty zestawu SDK. Dzięki temu, programista korzystający z Kinect for Windows SDK może z jednego miejsca mieć dostęp do dokumentacji, aktualizacji, przykładowych projektów i niezbędnych bibliotek.

Bardzo ciekawym i wszechstronnym narzędziem, do którego można uzyskać dostęp m.in. z poziomu Windows Developer Toolkit jest Kinect Studio. Kinect Studio jest aplikacją, którą podłącza się do działającego programu wykorzystującego sygnały płynące z czujnika Kinect. W ten sposób Kinect Studio uzyskuje dostęp do strumieni danych - strumienia głębokości oraz koloru, a także umożliwia podgląd trójwymiarowej przestrzeni w zasięgu pracy czujników. Występuje także funkcjonalność nagrywania sesji uruchomionego programu, dzięki czemu później można odtwarzać i analizować odbierane i przetwarzane sygnały i obrazy. Zrzut ekranu pokazujący działanie Kinect Studio zamieszczony jest poniżej.



Rysunek 2.7 Kinect Studio (źródło własne)

# Windows Presentation Foundation (WPF)

# Platforma .NET i język C#

# Implementacja systemu

Faza tworzenia systemu przeglądania dokumentacji medycznej podczas wykonywania procedur medycznych składała się z dwóch głównych elementów - projekt i implementacja wymaganych funkcjonalności oraz projekt i implementacja układu (ang. *layout*) interfejsu użytkownika programu.

## Wybór zastosowanych technologii

Skorzystano z technologii Windows Presentation Foundation, ponieważ umożliwia ona budowanie aplikacji komputerowych dedykowanych do obsługi materiałów multimedialnych. Co jednak było najistotniejsze przy wyborze tego rozwiązania, to fakt, iż WPF pozwalało na swobodną implementację interfejsu bezdotykowego. Często wykorzystywana technologia Windows Forms posiada interfejs przystosowany typowo do obsługi standardowymi urządzeniami wejściowymi - myszką i klawiaturą. Mnogość przycisków, rozwijane menu kontekstowe, układ okienek zdawały się być zbyt nieprzystępne do obsługi bezdotykowej. WPF z kolei udostępnia narzędzia do tworzenia oprogramowania także na urządzenia dotykowe (tablety, telefony). Co prawda możliwym jest zaimplementowanie obsługi kursora myszy za pomocą Kinect'a, jednak przy użyciu technologii Windows Forms byłoby to bardzo nieergonomiczne, nieintuicyjne i sprzeczne z naturalnym interfejsem użytkownika (NUI).

Mimo wspomnianych wyżej czynników, w jednym miejscu aplikacji wykorzystano standardową obsługę myszką - jest to pierwsze okno, które pokazuje się po uruchomieniu programu. Pozwala ono na wybór obrazów, które następnie użytkownik chce przeglądać za pomocą stworzonego systemu. W trakcie konsultacji z lekarzami ustalono, że inicjalizacja systemu przed samą procedurą medyczną, bez względu na późniejszy bezdotykowy interfejs, zachodzi przy pomocy standardowego sterowania myszką. Wynika to z faktu, iż dokumentacja danego pacjenta zapisana jest na nośniku CD i występuje konieczność jej skopiowania w celu przetwarzania. Inicjalizacja oprogramowania zostaje wykonana przez osobę, która przed rozpoczęciem wykonywania zabiegu lub operacji nie musi jeszcze zachować sterylności. Co więcej, wydaje się, że przeglądanie plików i folderów na komputerze obsługiwane bezdotykowo jest bardzo nieintuicyjnym i niezgrabnym rozwiązaniem.

Zdecydowano się na użycie języka programowania C# z trzech powodów:

* technologia WPF jest dostępna m.in. dla języka C#,
* biblioteki, przykładowe programy, kontrolki z zestawu SDK są dostępne w językach C++, C# oraz w technologii XNA,
* osobiste preferencje autorki.

W przypadku, gdyby przyjęte rozwiązanie okazało się zbyt wolne w działaniu, rozważano skorzystanie z języka C++, co jednak nie miało miejsca. W związku z tym, aplikację napisano przy użyciu technologii WPF w języku C#.

## Funkcjonalności

Wymaganiami co do funkcjonalności tworzonej aplikacji były: bezdotykowość oraz możliwości obrotu, przybliżania oraz oddalania wybranych zdjęć. Ponadto, dodano możliwość dodania kilku obrazów do programu i wyboru aktualnie przetwarzanego w trakcie korzystania z aplikacji. Aby zrealizować wymienione zagadnienia konieczne było zaimplementowanie rozpoznawania gestów. Czym zatem jest gest?

*"Gest jest ruchem ciała, który zawiera informacje. Machanie na pożegnanie jest gestem. Wciskanie klawisza na klawiaturze nie jest gestem, ponieważ ruch palca wykonywany w celu wciśnięcia przycisku nie jest ani obserwowany, ani istotny. To co jest ważne to fakt czy klawisz został wciśnięty."* (James Ashley, 2012)

Z przytoczonej definicji wynika, że tym co wyróżnia gest od innych ruchów i zachowań jest komunikat. Innymi słowy, aby mówić o wykonaniu gestu należy rozpoznać jego intencjonalność, uprzednią chęć wykonania go.

# Spis ilustracji

## Schematy

[Schemat 1.1 Sala operacyjna - widok z góry (opracowanie własne) 4](#_Toc396171291)

[Schemat 2.1 Interakcja sprzętu i oprogramowania z aplikacją (Microsoft Corporation, 2014) 6](#_Toc396171292)

[Schemat 2.2 Podzespoły urządzenia Kinect for Windows - opracowanie własne na podstawie (Microsoft Corporation, 2014) 7](#_Toc396171293)

[Schemat 2.3 Zakres wykrywania dźwięków (Microsoft Corporation, 2013) 8](#_Toc396171294)

[Schemat 2.4 Zasięg kątowy pracy czujnika głębokości i RGB (Microsoft Corporation, 2013) 8](#_Toc396171295)

[Schemat 2.5 Architektura SDK (Microsoft Corporation, 2014) 10](#_Toc396171296)

[Schemat 2.6 Układ współrzędnych dla strumienia głębokości (Microsoft Kinect, 2014) 16](#_Toc396171297)

[Schemat 2.7 Układ współrzędnych strumienia szkieletu (Microsoft Kinect, 2014) 19](#_Toc396171298)

[Schemat 2.8 Fizyczna strefa interakcji (ang. *Physical Interaction Zone*) (Microsoft Corporation, brak daty) 20](#_Toc396171299)

## Rysunki

[Rysunek 2.1 Obraz kolorowy (źródło własne) 14](#_Toc396171300)

[Rysunek 2.2 Obraz podczerwony (źródło własne) 15](#_Toc396171301)

[Rysunek 2.3 Obraz głębokości (źródło własne) 16](#_Toc396171302)

[Rysunek 2.4 Układ stawów - Skeleton Stream (Raiten, 2014) 17](#_Toc396171303)

[Rysunek 2.5 Tryb siedzący (Microsoft Corporation, brak daty) 17](#_Toc396171304)

[Rysunek 2.6 Śledzony szkielet (źródło własne) 19](#_Toc396171305)

[Rysunek 2.7 Kinect Studio (źródło własne) 22](#_Toc396171306)

# Bibliografia

Anon., 2014. *Developer Network.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh855356.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Anon., 2014. *Microsoft.* [Online]   
Available at: www.microsoft.com/en-us/kinectforwindows/meetkinect/default.aspx  
[Data uzyskania dostępu: 17 08 2014].

Anon., 2014. *Natural user interface - Wikipedia.* [Online]   
Available at: http://en.wikipedia.org/wiki/Natural\_user\_interface  
[Data uzyskania dostępu: 17 sierpień 2014].

Anon., brak daty *Developer Network.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131033.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Anon., brak daty *Developer Network.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj663790.aspx

Anon., brak daty *Developer Network.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131023.aspx

Azimi, M., 2014. *Skeletal Joint Smoothing White Paper,* brak miejsca: brak nazwiska

Catuhe, D., 2012. *Programming with the Kinect for Windows Software Development Kit.* Redmond, Washington: Microsoft Press.

Corish Bob, C. A. ,. G. G. ,. O. K. S. A., 2014. *Touchless Interaction in Medical Imaging.* [Online]   
Available at: http://research.microsoft.com/en-us/projects/touchlessinteractionmedical/  
[Data uzyskania dostępu: 17 sieprpień 2014].

James Ashley, J. W., 2012. *Beginning Kinect Programming with the Microsoft Kinect SDK.* New York: Apress.

Microsft Corporation, brak daty *Microsoft Kinect for Windows Software Development Kit (SDK) End User License Agreement.* brak miejsca:brak nazwiska

Microsoft Corporation, 2013. *Human Interface Guidelines v1.8.* brak miejsca:brak nazwiska

Microsoft Corporation, 2014. *Accelerometer.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj663790.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Color Stream.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131027.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Depth Stream.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131028.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *DirectX Media Objects.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/windows/desktop/dd375474%28v=vs.85%29.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Getting the Next Frame of Data by Polling or Using Events.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh973076.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Joint Filtering.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131024.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Kinect for Windows - Meet Kinect.* [Online]   
Available at: www.microsoft.com/en-us/kinectforwindows/meetkinect/default.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Kinect for Windows Architecture.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131023.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Kinect for Windows Sensor Components and Specifications.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/jj131033.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *KinectInteraction Architecture.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/dn188672.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Running a Kinect-enabled Application on a Developer Machine.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh855358.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *Setting Up a Kinect Sensor.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh855356.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, 2014. *System Requirements.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh855359.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Microsoft Corporation, brak daty *Human Interface Guidelines v1.8.* brak miejsca:brak nazwiska

Microsoft Kinect, 2014. *Coordinate Spaces.* [Online]   
Available at: http://msdn.microsoft.com/en-us/library/hh973078.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

Raiten, S., 2014. *Kinect – Getting Started – Become The Incredible Hulk.* [Online]   
Available at: http://blogs.microsoft.co.il/blogs/shair/image\_48A9C26C.png  
[Data uzyskania dostępu: 17 sierpień 2014].

Stango, N., 2012. *Kinect Can Help Doctors Monitor Your Physical Rehab From Home.* [Online]   
Available at: http://gizmodo.com/5946737/kinect-can-doctors-monitor-your-physical-rehab-from-home  
[Data uzyskania dostępu: 17 sierpień 2014].

The Kinect for Windows Team, 2014. *Jintronix makes rehabilitation more convenient, fun, and affordable with Kinect for Windows.* [Online]   
Available at: http://blogs.msdn.com/b/kinectforwindows/archive/2014/01/22/jintronix-makes-rehabilitation-more-convenient-fun-and-affordable-with-kinect-for-windows.aspx  
[Data uzyskania dostępu: 17 sierpień 2014].

1. DMO - komponent strumieniowania danych oparty o technologię COM (tłumaczenie własne) [↑](#footnote-ref-1)