

SensorGloveQt

1.0

Wygenerowano przez Doxygen 1.8.13

Spis treści

1	Indeks przestrzeni nazw	1
1.1	Lista przestrzeni nazw	1
2	Indeks hierarchiczny	3
2.1	Hierarchia klas	3
3	Indeks klas	5
3.1	Lista klas	5
4	Indeks plików	7
4.1	Lista plików	7
5	Dokumentacja przestrzeni nazw	9
5.1	Dokumentacja przestrzeni nazw Ui	9
6	Dokumentacja klas	11
6.1	Dokumentacja klasy ErrorHandler	11
6.1.1	Opis szczegółowy	11
6.1.2	Dokumentacja składowych wyliczanych	11
6.1.2.1	ErrorType_t	11
6.1.3	Dokumentacja konstruktora i destruktor	12
6.1.3.1	ErrorHandler()	12
6.1.4	Dokumentacja funkcji składowych	12
6.1.4.1	ExceptionHandler()	12
6.2	Dokumentacja klasy Finger	12
6.2.1	Opis szczegółowy	13

6.2.2	Dokumentacja składowych wyliczanych	13
6.2.2.1	FingerType_t	13
6.2.3	Dokumentacja konstruktora i destruktora	13
6.2.3.1	Finger() [1/2]	14
6.2.3.2	Finger() [2/2]	14
6.2.4	Dokumentacja funkcji składowych	14
6.2.4.1	GetFingertipValue()	14
6.2.4.2	operator=()	14
6.2.4.3	SetFingertipValue()	14
6.2.4.4	TransformJointAngles()	15
6.2.5	Dokumentacja atrybutów składowych	15
6.2.5.1	m_fingertip	15
6.2.5.2	m_fingerType	15
6.2.5.3	m_manipulator	15
6.2.5.4	m_rootEntity	15
6.3	Dokumentacja klasy Fingertip	16
6.3.1	Opis szczegółowy	16
6.3.2	Dokumentacja konstruktora i destruktora	16
6.3.2.1	Fingertip()	16
6.3.3	Dokumentacja funkcji składowych	17
6.3.3.1	getFingertipValue()	17
6.3.3.2	setFingertipValue()	17
6.3.3.3	TransformFingertip()	17
6.3.4	Dokumentacja atrybutów składowych	17
6.3.4.1	m_fingertipValue	17
6.3.4.2	m_rootEntity	17
6.3.4.3	m_sphereEntity	18
6.3.4.4	m_sphereMaterial	18
6.3.4.5	m_sphereTransform	18
6.3.4.6	m_TransformMatrix	18

6.4	Dokumentacja klasy Hand	18
6.4.1	Opis szczegółowy	19
6.4.2	Dokumentacja konstruktora i destruktora	19
6.4.2.1	Hand()	19
6.4.3	Dokumentacja funkcji składowych	19
6.4.3.1	GetFingertipValues()	19
6.4.3.2	SetFingerJoints()	20
6.4.3.3	SetFingertipValues()	20
6.4.4	Dokumentacja atrybutów składowych	20
6.4.4.1	m_fingers	20
6.4.4.2	m_rootEntity	20
6.5	Dokumentacja klasy Input	20
6.5.1	Opis szczegółowy	21
6.5.2	Dokumentacja składowych wyliczanych	21
6.5.2.1	ConnectionType_t	21
6.5.3	Dokumentacja konstruktora i destruktora	21
6.5.3.1	Input()	22
6.5.4	Dokumentacja funkcji składowych	22
6.5.4.1	ChangeConnectionType()	22
6.5.4.2	ConfigureInput()	22
6.5.5	Dokumentacja atrybutów składowych	22
6.5.5.1	m_AccelerometerValues	22
6.5.5.2	m_ConnectionType	22
6.5.5.3	m_JointAngles	23
6.5.5.4	m_TensionSensorValues	23
6.6	Dokumentacja klasy Joint	23
6.6.1	Opis szczegółowy	24
6.6.2	Dokumentacja konstruktora i destruktora	24
6.6.2.1	Joint() [1/2]	24
6.6.2.2	Joint() [2/2]	24

6.6.3	Dokumentacja funkcji składowych	24
6.6.3.1	angle()	25
6.6.3.2	length()	25
6.6.3.3	operator=()	25
6.6.3.4	TransformAngle()	25
6.6.3.5	TransformMatrix()	25
6.6.4	Dokumentacja atrybutów składowych	25
6.6.4.1	m_angle	26
6.6.4.2	m_CylinderEntity	26
6.6.4.3	m_CylinderTransform	26
6.6.4.4	m_length	26
6.6.4.5	m_position	26
6.6.4.6	m_radius	26
6.6.4.7	m_rootEntity	27
6.6.4.8	m_rotation	27
6.6.4.9	m_tilt	27
6.6.4.10	m_TransformMatrix	27
6.7	Dokumentacja klasy MainWindow	27
6.7.1	Opis szczegółowy	28
6.7.2	Dokumentacja konstruktora i destruktor	28
6.7.2.1	MainWindow()	28
6.7.2.2	~MainWindow()	28
6.7.3	Dokumentacja funkcji składowych	29
6.7.3.1	AddWidgetToGlove3DLayout()	29
6.7.3.2	InitInputData()	29
6.7.3.3	on_CameraOrientationLineEdit_editingFinished	29
6.7.3.4	on_CameraOrientationLineEdit_textEdited	29
6.7.3.5	on_CameraOrientationSlider_valueChanged	29
6.7.3.6	on_GloveZoomLineEdit_editingFinished	30
6.7.3.7	on_GloveZoomLineEdit_textEdited	30

6.7.3.8	on_GloveZoomSlider_valueChanged	30
6.7.3.9	on_StartStopButton_clicked	30
6.7.4	Dokumentacja atrybutów składowych	30
6.7.4.1	InputData	30
6.7.4.2	ui	31
6.8	Dokumentacja klasy ManipulatorRotational	31
6.8.1	Opis szczegółowy	31
6.8.2	Dokumentacja konstruktora i destruktora	32
6.8.2.1	ManipulatorRotational() [1/2]	32
6.8.2.2	ManipulatorRotational() [2/2]	32
6.8.3	Dokumentacja funkcji składowych	32
6.8.3.1	getLastJoint()	32
6.8.3.2	TransformJointAngles()	32
6.8.4	Dokumentacja atrybutów składowych	32
6.8.4.1	m_joints	33
6.8.4.2	m_position	33
6.8.4.3	m_rootEntity	33
6.8.4.4	m_rotation	33
6.9	Dokumentacja klasy Scene	33
6.9.1	Opis szczegółowy	34
6.9.2	Dokumentacja konstruktora i destruktora	34
6.9.2.1	Scene()	34
6.9.2.2	~Scene()	35
6.9.3	Dokumentacja funkcji składowych	35
6.9.3.1	SetHandFingertipValues()	35
6.9.3.2	SetHandTransformation()	35
6.9.4	Dokumentacja atrybutów składowych	35
6.9.4.1	camController	35
6.9.4.2	cameraEntity	35
6.9.4.3	container	36

6.9.4.4	hLayout	36
6.9.4.5	input	36
6.9.4.6	light	36
6.9.4.7	light2	36
6.9.4.8	lightEntity	36
6.9.4.9	lightEntity2	37
6.9.4.10	lightTransform	37
6.9.4.11	lightTransform2	37
6.9.4.12	m_Hand3DModel	37
6.9.4.13	m_rootEntity	37
6.9.4.14	screenSize	37
6.9.4.15	view	38
6.9.4.16	vLayout	38
6.9.4.17	widget	38
7	Dokumentacja plików	39
7.1	Dokumentacja pliku array.cpp	39
7.2	Dokumentacja pliku errorHandler.cpp	39
7.3	Dokumentacja pliku errorHandler.h	39
7.3.1	Opis szczegółowy	39
7.4	Dokumentacja pliku finger.cpp	39
7.5	Dokumentacja pliku finger.h	40
7.5.1	Opis szczegółowy	40
7.5.2	Dokumentacja definicji	41
7.5.2.1	FINGER_JOINTS_COUNT	41
7.5.2.2	FINGER_RADIUS	41
7.5.2.3	INDEX_FINGER_JOINT0_LENGTH	41
7.5.2.4	INDEX_FINGER_JOINT1_LENGTH	41
7.5.2.5	INDEX_FINGER_JOINT2_LENGTH	41
7.5.2.6	INDEX_FINGER_JOINT3_LENGTH	42
7.5.2.7	INDEX_JOINT_COUNT	42

7.5.2.8	MIDDLE_FINGER_JOINT0_LENGTH	42
7.5.2.9	MIDDLE_FINGER_JOINT1_LENGTH	42
7.5.2.10	MIDDLE_FINGER_JOINT2_LENGTH	42
7.5.2.11	MIDDLE_FINGER_JOINT3_LENGTH	42
7.5.2.12	MIDDLE_JOINT_COUNT	43
7.5.2.13	PINKY_JOINT0_LENGTH	43
7.5.2.14	PINKY_JOINT1_LENGTH	43
7.5.2.15	PINKY_JOINT2_LENGTH	43
7.5.2.16	PINKY_JOINT3_LENGTH	43
7.5.2.17	PINKY_JOINT_COUNT	43
7.5.2.18	RING_FINGER_JOINT0_LENGTH	44
7.5.2.19	RING_FINGER_JOINT1_LENGTH	44
7.5.2.20	RING_FINGER_JOINT2_LENGTH	44
7.5.2.21	RING_FINGER_JOINT3_LENGTH	44
7.5.2.22	RING_JOINT_COUNT	44
7.5.2.23	THUMB_JOINT0_LENGTH	44
7.5.2.24	THUMB_JOINT1_LENGTH	45
7.5.2.25	THUMB_JOINT2_LENGTH	45
7.5.2.26	THUMB_JOINT_COUNT	45
7.6	Dokumentacja pliku fingertip.cpp	45
7.7	Dokumentacja pliku fingertip.h	45
7.7.1	Opis szczegółowy	46
7.7.2	Dokumentacja definicji	46
7.7.2.1	SPHERE_RADIUS	46
7.7.2.2	SPHERE_RINGS_COUNT	46
7.7.2.3	SPHERE_SCALE	46
7.7.2.4	SPHERE_SLICES_COUNT	46
7.8	Dokumentacja pliku hand.cpp	47
7.9	Dokumentacja pliku hand.h	47
7.9.1	Opis szczegółowy	47

7.9.2	Dokumentacja definicji	47
7.9.2.1	FINGER_COUNT	48
7.9.2.2	INDEX_FINGER_OFFSET	48
7.9.2.3	INDEX_FINGER_ROTATION	48
7.9.2.4	MIDDLE_FINGER_OFFSET	48
7.9.2.5	MIDDLE_FINGER_ROTATION	48
7.9.2.6	PINKY_FINGER_OFFSET	48
7.9.2.7	PINKY_FINGER_ROTATION	49
7.9.2.8	RING_FINGER_OFFSET	49
7.9.2.9	RING_FINGER_ROTATION	49
7.9.2.10	THUMB_FINGER_OFFSET	49
7.9.2.11	THUMB_FINGER_ROTATION	49
7.10	Dokumentacja pliku includes.h	50
7.10.1	Opis szczegółowy	50
7.11	Dokumentacja pliku input.cpp	51
7.12	Dokumentacja pliku input.h	51
7.12.1	Opis szczegółowy	51
7.13	Dokumentacja pliku joint.cpp	51
7.14	Dokumentacja pliku joint.h	51
7.14.1	Opis szczegółowy	52
7.14.2	Dokumentacja definicji	52
7.14.2.1	CYLINDER_RINGS_COUNT	52
7.14.2.2	CYLINDER_SLICES_COUNT	52
7.15	Dokumentacja pliku main.cpp	52
7.15.1	Dokumentacja funkcji	52
7.15.1.1	main()	53
7.16	Dokumentacja pliku mainwindow.cpp	53
7.17	Dokumentacja pliku mainwindow.h	53
7.17.1	Opis szczegółowy	53
7.18	Dokumentacja pliku manipulatorrotational.cpp	53
7.19	Dokumentacja pliku manipulatorrotational.h	54
7.19.1	Opis szczegółowy	54
7.20	Dokumentacja pliku scene.cpp	54
7.21	Dokumentacja pliku scene.h	54
7.21.1	Opis szczegółowy	54

Rozdział 1

Indeks przestrzeni nazw

1.1 Lista przestrzeni nazw

Tutaj znajdują się wszystkie przestrzenie nazw wraz z ich krótkimi opisami:

Ui	9
----	---

Rozdział 2

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

ErrorHandler	11
QBluetoothTransferManager	
Input	20
QMainWindow	
MainWindow	27
QObject	
Finger	12
Fingertip	16
Hand	18
Joint	23
ManipulatorRotational	31
Scene	33
QSerialPort	
Input	20

Rozdział 3

Indeks klas

3.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

ErrorHandler	Klasa obsługująca błędy i wyjątki	11
Finger	Modeluje pojęcie palca w przestrzeni trójwymiarowej	12
Fingertip	Opisuje opuszki palców	16
Hand	Modeluje pojęcie dłoni	18
Input	Klasa jest narzędziem do pobierania danych	20
Joint	Modeluje pojęcie stawu	23
MainWindow	Modeluje strukturę GUI programu	27
ManipulatorRotational	Definiuje pojęcie manipulatora rotacyjnego	31
Scene	Modeluje pojęcie sceny z elementami 3D	33

Rozdział 4

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

array.cpp	39
errorhandler.cpp	39
errorhandler.h	
Definicja klasy ErrorHandler	39
finger.cpp	39
finger.h	
Definicja klasy kwadrat	40
fingertip.cpp	45
fingertip.h	
Definicja klasy Fingertip	45
hand.cpp	47
hand.h	
Definicja klasy Hand	47
includes.h	
Dołączenie bibliotek	50
input.cpp	51
input.h	
Definicja klasy Input	51
joint.cpp	51
joint.h	
Definicja klasy Joint	51
main.cpp	52
mainwindow.cpp	53
mainwindow.h	
Definicja klasy MainWindow	53
manipulatorrotational.cpp	53
manipulatorrotational.h	
Definicja klasy ManipulatorRotational	54
scene.cpp	54
scene.h	
Definicja klasy Scene	54

Rozdział 5

Dokumentacja przestrzeni nazw

5.1 Dokumentacja przestrzeni nazw Ui

Rozdział 6

Dokumentacja klas

6.1 Dokumentacja klasy ErrorHandler

Klasa obsługująca błędy i wyjątki.

```
#include <errorhandler.h>
```

Typy publiczne

- enum `ErrorType_t` { `NegativeAngle` =0 }

Metody publiczne

- `ErrorHandler` ()
- void `ExceptionHandler` (`ErrorType_t` p_errorType)

6.1.1 Opis szczegółowy

Definicja w linii 16 pliku errorhandler.h.

6.1.2 Dokumentacja składowych wyliczanych

6.1.2.1 ErrorType_t

```
enum ErrorHandler::ErrorType_t
```

Wartości wyliczeń

NegativeAngle

Definicja w linii 19 pliku `errorhandler.h`.

6.1.3 Dokumentacja konstruktora i destruktor

6.1.3.1 ErrorHandler()

```
ErrorHandler::ErrorHandler ( )
```

Definicja w linii 3 pliku `errorhandler.cpp`.

6.1.4 Dokumentacja funkcji składowych

6.1.4.1 ExceptionHandler()

```
void ErrorHandler::ExceptionHandler (
    ErrorHandler::ErrorType_t p_errorType )
```

Definicja w linii 8 pliku `errorhandler.cpp`.

Dokumentacja dla tej klasy została wygenerowana z plików:

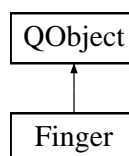
- [errorhandler.h](#)
- [errorhandler.cpp](#)

6.2 Dokumentacja klasy Finger

Modeluje pojęcie palca w przestrzeni trójwymiarowej.

```
#include <finger.h>
```

Diagram dziedziczenia dla Finger



Typy publiczne

- enum `FingerType_t` {
 `Thumb`, `IndexFinger`, `MiddleFinger`, `RingFinger`,
 `Pinky` }

Metody publiczne

- `Finger` (`Qt3DCore::QEntity *p_rootEntity=nullptr`, `FingerType_t p_fingerType=IndexFinger`, `QVector3D p_position=QVector3D()`, `float p_rotation=float()`, `QColor p_fingerColor=QColor(255, 173, 96)`)
- `Finger` (`const Finger &p_finger`)
- `Finger &operator=` (`const Finger &p_finger`)
- `void TransformJointAngles` (`QVector< float > p_angles`)
- `int GetFingertipValue` () `const`
- `void SetFingertipValue` (`int p_NewFingertipValue`)

Atrybuty chronione

- `Fingertip * m_fingertip`
- `Qt3DCore::QEntity * m_rootEntity`
- `ManipulatorRotational * m_manipulator`
- `FingerType_t m_fingerType`

6.2.1 Opis szczegółowy

Inicjalizuje identyfikator właściwy dla danego typu palca, położenie w przestrzeni 3D, kąt nachylenia i kolor trójwymiarowego modelu.

Definicja w linii 62 pliku `finger.h`.

6.2.2 Dokumentacja składowych wyliczanych

6.2.2.1 FingerType_t

```
enum Finger::FingerType_t
```

Wartości wyliczeń

Thumb	
IndexFinger	
MiddleFinger	
RingFinger	
Pinky	

Definicja w linii 66 pliku `finger.h`.

6.2.3 Dokumentacja konstruktora i destruktor

6.2.3.1 Finger() [1/2]

```
Finger::Finger (
    Qt3DCore::QEntity * p_rootEntity = nullptr,
    FingerType_t p_fingerType = IndexFinger,
    QVector3D p_position = QVector3D(),
    float p_rotation = float(),
    QColor p_fingerColor = QColor(255,173,96) )
```

Definicja w linii 3 pliku finger.cpp.

6.2.3.2 Finger() [2/2]

```
Finger::Finger (
    const Finger & p_finger )
```

Definicja w linii 80 pliku finger.cpp.

6.2.4 Dokumentacja funkcji składowych

6.2.4.1 GetFingertipValue()

```
int Finger::GetFingertipValue ( ) const
```

Definicja w linii 113 pliku finger.cpp.

6.2.4.2 operator=()

```
Finger & Finger::operator= (
    const Finger & p_finger )
```

Definicja w linii 88 pliku finger.cpp.

6.2.4.3 SetFingertipValue()

```
void Finger::SetFingertipValue (
    int p_NewFingertipValue )
```

Definicja w linii 118 pliku finger.cpp.

6.2.4.4 TransformJointAngles()

```
void Finger::TransformJointAngles (
    QVector< float > p_angles )
```

Definicja w linii 97 pliku finger.cpp.

6.2.5 Dokumentacja atrybutów składowych

6.2.5.1 m_fingertip

```
Fingertip* Finger::m_fingertip [protected]
```

Definicja w linii 68 pliku finger.h.

6.2.5.2 m_fingerType

```
FingerType_t Finger::m_fingerType [protected]
```

Definicja w linii 73 pliku finger.h.

6.2.5.3 m_manipulator

```
ManipulatorRotational* Finger::m_manipulator [protected]
```

Definicja w linii 71 pliku finger.h.

6.2.5.4 m_rootEntity

```
Qt3DCore::QEntity* Finger::m_rootEntity [protected]
```

Definicja w linii 70 pliku finger.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

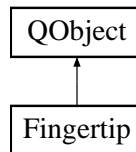
- [finger.h](#)
- [finger.cpp](#)

6.3 Dokumentacja klasy Fingertip

Opisuje opuszki palców.

```
#include <fingertip.h>
```

Diagram dziedziczenia dla Fingertip



Metody publiczne

- `Fingertip` (`Qt3DCore::QEntity *p_rootEntity=nullptr`, `Joint *p_PrevJoint=nullptr`, `int p_fingertipValue=int()`)
- `void TransformFingertip (Joint *p_prevJoint)`
- `void setFingertipValue (int p_newValue)`
- `float getFingertipValue () const`

Atrybuty chronione

- `int m_fingertipValue`
- `Qt3DCore::QEntity * m_rootEntity`
- `Qt3DCore::QEntity * m_sphereEntity`
- `Qt3DCore::QTransform * m_sphereTransform`
- `Qt3DExtras::QPhongMaterial * m_sphereMaterial`
- `QMatrix4x4 m_TransformMatrix`

6.3.1 Opis szczegółowy

Reprezentuje trójwymiarową wizualizację opuszków palców w postaci obiektów sferycznych. Służy do wizualizacji odczytów z czujników nacisku umieszczonych na opuszkach palców rękawicy sensorycznej. Inicjalizuje rozmiar sfery i położenie w przedtrzeni trówymiarowej.

Definicja w linii 33 pliku `fingertip.h`.

6.3.2 Dokumentacja konstruktora i destruktora

6.3.2.1 Fingertip()

```

Fingertip::Fingertip (
    Qt3DCore::QEntity * p_rootEntity = nullptr,
    Joint * p_PrevJoint = nullptr,
    int p_fingertipValue = int() ) [explicit]
  
```

Definicja w linii 3 pliku `fingertip.cpp`.

6.3.3 Dokumentacja funkcji składowych

6.3.3.1 getFingertipValue()

```
float Fingertip::getFingertipValue ( ) const
```

Definicja w linii 49 pliku fingertip.cpp.

6.3.3.2 setFingertipValue()

```
void Fingertip::setFingertipValue (
    int p_newValue )
```

Definicja w linii 40 pliku fingertip.cpp.

6.3.3.3 TransformFingertip()

```
void Fingertip::TransformFingertip (
    Joint * p_prevJoint )
```

Definicja w linii 33 pliku fingertip.cpp.

6.3.4 Dokumentacja atrybutów składowych

6.3.4.1 m_fingertipValue

```
int Fingertip::m_fingertipValue [protected]
```

Definicja w linii 37 pliku fingertip.h.

6.3.4.2 m_rootEntity

```
Qt3DCore::QEntity* Fingertip::m_rootEntity [protected]
```

Definicja w linii 38 pliku fingertip.h.

6.3.4.3 m_sphereEntity

```
Qt3DCore::QEntity* Fingertip::m_sphereEntity [protected]
```

Definicja w linii 39 pliku fingertip.h.

6.3.4.4 m_sphereMaterial

```
Qt3DExtras::QPhongMaterial* Fingertip::m_sphereMaterial [protected]
```

Definicja w linii 41 pliku fingertip.h.

6.3.4.5 m_sphereTransform

```
Qt3DCore::QTransform* Fingertip::m_sphereTransform [protected]
```

Definicja w linii 40 pliku fingertip.h.

6.3.4.6 m_TransformMatrix

```
QMatrix4x4 Fingertip::m_TransformMatrix [protected]
```

Definicja w linii 42 pliku fingertip.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

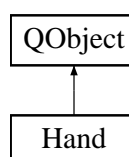
- [fingertip.h](#)
- [fingertip.cpp](#)

6.4 Dokumentacja klasy Hand

Modeluje pojęcie dłoni.

```
#include <hand.h>
```

Diagram dziedziczenia dla Hand



Metody publiczne

- [Hand](#) (Qt3DCore::QEntity *p_rootEntity=nullptr, QColor p_HandColor=QColor(255, 173, 96))
- QVector< int > [GetFingertipValues](#) () const
- void [SetFingerJoints](#) (QVector< QVector< float >> p_JointAngles)
- void [SetFingertipValues](#) (QVector< int > p_FingertipValues)

Atrybuty chronione

- Qt3DCore::QEntity * [m_rootEntity](#)
- QVector< [Finger](#) > [m_fingers](#)

6.4.1 Opis szczegółowy

Klasa składa się z elementów palców, rozmieszczonych w przestrzeni trójwymiarowej

Definicja w linii 39 pliku hand.h.

6.4.2 Dokumentacja konstruktora i destruktor

6.4.2.1 Hand()

```
Hand::Hand (
    Qt3DCore::QEntity * p_rootEntity = nullptr,
    QColor p_HandColor = QColor(255,173,96) )
```

Definicja w linii 3 pliku hand.cpp.

6.4.3 Dokumentacja funkcji składowych

6.4.3.1 GetFingertipValues()

```
QVector< int > Hand::GetFingertipValues ( ) const
```

Definicja w linii 27 pliku hand.cpp.

6.4.3.2 SetFingerJoints()

```
void Hand::SetFingerJoints (
    QVector< QVector< float >> p_JointAngles )
```

Definicja w linii 35 pliku hand.cpp.

6.4.3.3 SetFingertipValues()

```
void Hand::SetFingertipValues (
    QVector< int > p_FingertipValues )
```

Definicja w linii 43 pliku hand.cpp.

6.4.4 Dokumentacja atrybutów składowych

6.4.4.1 m_fingers

```
QVector<Finger> Hand::m_fingers [protected]
```

Definicja w linii 44 pliku hand.h.

6.4.4.2 m_rootEntity

```
Qt3DCore::QEntity* Hand::m_rootEntity [protected]
```

Definicja w linii 43 pliku hand.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

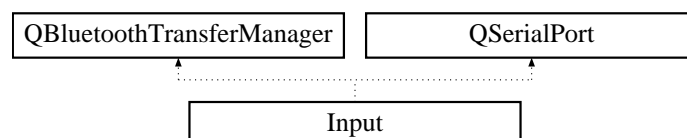
- [hand.h](#)
- [hand.cpp](#)

6.5 Dokumentacja klasy Input

Klasa jest narzędziem do pobierania danych.

```
#include <input.h>
```

Diagram dziedziczenia dla Input



Metody publiczne

- [Input](#) ()
- void [ChangeConnectionType](#) ([ConnectionType_t](#) p_ConnecitonType)
- void [ConfigureInput](#) ()

Typy prywatne

- enum [ConnectionType_t](#) { [BluetoothConnection](#), [USBConnection](#) }

Atrybuty prywatne

- [QVector< float > m_JointAngles](#)
- [QVector< int > m_TensionSensorValues](#)
- [QVector< float > m_AccelerometerValues](#)
- [ConnectionType_t m_ConnectionType](#)

6.5.1 Opis szczegółowy

Inicjalizuje kąty między stawami, wskazania czujników tensometrycznych, wskazania akcelerometru i typ połączenia z rękawicą sensoryczną

Definicja w linii 29 pliku input.h.

6.5.2 Dokumentacja składowych wyliczanych

6.5.2.1 ConnectionType_t

```
enum Input::ConnectionType\_t [private]
```

Wartości wyliczeń

BluetoothConnection	
USBConnection	

Definicja w linii 31 pliku input.h.

6.5.3 Dokumentacja konstruktora i destruktor

6.5.3.1 Input()

```
Input::Input ( )
```

Definicja w linii 3 pliku input.cpp.

6.5.4 Dokumentacja funkcji składowych

6.5.4.1 ChangeConnectionType()

```
void Input::ChangeConnectionType (
    Input::ConnectionType_t p_ConnecitonType )
```

Definicja w linii 8 pliku input.cpp.

6.5.4.2 ConfigureInput()

```
void Input::ConfigureInput ( )
```

6.5.5 Dokumentacja atrybutów składowych

6.5.5.1 m_AccelerometerValues

```
QVector<float> Input::m_AccelerometerValues [private]
```

Definicja w linii 35 pliku input.h.

6.5.5.2 m_ConnectionType

```
ConnectionType_t Input::m_ConnectionType [private]
```

Definicja w linii 36 pliku input.h.

6.5.5.3 m_JointAngles

```
QVector<float> Input::m_JointAngles [private]
```

Definicja w linii 33 pliku input.h.

6.5.5.4 m_TensionSensorValues

```
QVector<int> Input::m_TensionSensorValues [private]
```

Definicja w linii 34 pliku input.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

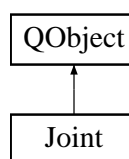
- [input.h](#)
- [input.cpp](#)

6.6 Dokumentacja klasy Joint

Modeluje pojęcie stawu.

```
#include <joint.h>
```

Diagram dziedziczenia dla Joint



Metody publiczne

- [Joint](#) (Qt3DCore::QEntity *p_rootEntity=nullptr, float p_radius=1, float p_length=1, float p_angle=0, float p_tilt=0, float p_rotation=0, QVector3D p_position=QVector3D(), [Joint](#) *p_PrevJoint=nullptr, QColor p_color=QColor(255, 173, 96))
- [Joint](#) (const [Joint](#) &p_joint)
- float [length](#) () const
- float [angle](#) () const
- QMatrix4x4 [TransformMatrix](#) () const
- void [TransformAngle](#) (float p_newAngle, [Joint](#) *p_PrevJoint=nullptr)
- [Joint](#) & [operator=](#) (const [Joint](#) &p_joint)

Atrybuty chronione

- float `m_radius`
- float `m_length`
- float `m_angle`
- float `m_tilt`
- float `m_rotation`
- QVector3D `m_position`
- Qt3DCore::QEntity * `m_rootEntity`
- Qt3DCore::QEntity * `m_CylinderEntity`
- Qt3DCore::QTransform * `m_CylinderTransform`
- QMatrix4x4 `m_TransformMatrix`

6.6.1 Opis szczegółowy

Inicjalizuje długość stawu, kąt nachylenia względem poprzedniego stawu

Definicja w linii 27 pliku joint.h.

6.6.2 Dokumentacja konstruktora i destruktor

6.6.2.1 Joint() [1/2]

```
Joint::Joint (
    Qt3DCore::QEntity * p_rootEntity = nullptr,
    float p_radius = 1,
    float p_length = 1,
    float p_angle = 0,
    float p_tilt = 0,
    float p_rotation = 0,
    QVector3D p_position = QVector3D(),
    Joint * p_PrevJoint = nullptr,
    QColor p_color = QColor(255,173,96) )
```

Definicja w linii 3 pliku joint.cpp.

6.6.2.2 Joint() [2/2]

```
Joint::Joint (
    const Joint & p_joint )
```

Definicja w linii 44 pliku joint.cpp.

6.6.3 Dokumentacja funkcji składowych

6.6.3.1 angle()

```
float Joint::angle ( ) const
```

Definicja w linii 63 pliku joint.cpp.

6.6.3.2 length()

```
float Joint::length ( ) const
```

Definicja w linii 58 pliku joint.cpp.

6.6.3.3 operator=()

```
Joint & Joint::operator= (
    const Joint & p_joint )
```

Definicja w linii 94 pliku joint.cpp.

6.6.3.4 TransformAngle()

```
void Joint::TransformAngle (
    float p_newAngle,
    Joint * p_PrevJoint = nullptr )
```

Definicja w linii 73 pliku joint.cpp.

6.6.3.5 TransformMatrix()

```
QMatrix4x4 Joint::TransformMatrix ( ) const
```

Definicja w linii 68 pliku joint.cpp.

6.6.4 Dokumentacja atrybutów składowych

6.6.4.1 m_angle

```
float Joint::m_angle [protected]
```

Definicja w linii 33 pliku joint.h.

6.6.4.2 m_CylinderEntity

```
Qt3DCore::QEntity* Joint::m_CylinderEntity [protected]
```

Definicja w linii 38 pliku joint.h.

6.6.4.3 m_CylinderTransform

```
Qt3DCore::QTransform* Joint::m_CylinderTransform [protected]
```

Definicja w linii 39 pliku joint.h.

6.6.4.4 m_length

```
float Joint::m_length [protected]
```

Definicja w linii 32 pliku joint.h.

6.6.4.5 m_position

```
QVector3D Joint::m_position [protected]
```

Definicja w linii 36 pliku joint.h.

6.6.4.6 m_radius

```
float Joint::m_radius [protected]
```

Definicja w linii 31 pliku joint.h.

6.6.4.7 m_rootEntity

```
Qt3DCore::QEntity* Joint::m_rootEntity [protected]
```

Definicja w linii 37 pliku joint.h.

6.6.4.8 m_rotation

```
float Joint::m_rotation [protected]
```

Definicja w linii 35 pliku joint.h.

6.6.4.9 m_tilt

```
float Joint::m_tilt [protected]
```

Definicja w linii 34 pliku joint.h.

6.6.4.10 m_TransformMatrix

```
QMatrix4x4 Joint::m_TransformMatrix [protected]
```

Definicja w linii 40 pliku joint.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

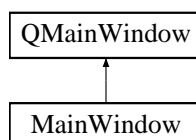
- [joint.h](#)
- [joint.cpp](#)

6.7 Dokumentacja klasy MainWindow

Modeluje strukturę GUI programu.

```
#include <mainwindow.h>
```

Diagram dziedziczenia dla MainWindow



Metody publiczne

- [MainWindow](#) (QWidget *parent=0)
- [~MainWindow](#) ()
- void [AddWidgetToGlove3DLayout](#) (QWidget *widget)
- void [InitInputData](#) (Input *data)

Sloty prywatne

- void [on_StartStopButton_clicked](#) ()
- void [on_CameraOrientationSlider_valueChanged](#) (int value)
- void [on_CameraOrientationLineEdit_textEdited](#) (const QString &arg1)
- void [on_CameraOrientationLineEdit_editingFinished](#) ()
- void [on_GloveZoomSlider_valueChanged](#) (int value)
- void [on_GloveZoomLineEdit_textEdited](#) (const QString &arg1)
- void [on_GloveZoomLineEdit_editingFinished](#) ()

Atrybuty prywatne

- Ui::MainWindow * [ui](#)
- [Input](#) * [InputData](#)

6.7.1 Opis szczegółowy

Inicjalizuje główne okno programu, jego elementy składowe w postaci widgetów oraz wczytuje zewnętrzny widget trójwymiarowej wizualizacji dłoni. Klasa pobiera dane z obiektu typu [Input](#), aktualizuje dane w modelu dłoni typu [Hand](#) i na ich podstawie wyświetla aktualne informacje dotyczące położenia stawów, nacisku i wskazania akcelero-metru. Pozwala na zmianę perspektywy widoku na model trójwymiarowy, uruchomienie i zatrzymanie pomiarów

Definicja w linii 34 pliku mainwindow.h.

6.7.2 Dokumentacja konstruktora i destruktor

6.7.2.1 MainWindow()

```
MainWindow::MainWindow (  
    QWidget * parent = 0 ) [explicit]
```

Definicja w linii 3 pliku mainwindow.cpp.

6.7.2.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

Definicja w linii 11 pliku mainwindow.cpp.

6.7.3 Dokumentacja funkcji składowych

6.7.3.1 AddWidgetToGlove3DLayout()

```
void MainWindow::AddWidgetToGlove3DLayout (
    QWidget * widget )
```

Definicja w linii 16 pliku mainwindow.cpp.

6.7.3.2 InitInputData()

```
void MainWindow::InitInputData (
    Input * data )
```

Definicja w linii 23 pliku mainwindow.cpp.

6.7.3.3 on_CameraOrientationLineEdit_editingFinished

```
void MainWindow::on_CameraOrientationLineEdit_editingFinished ( ) [private], [slot]
```

Definicja w linii 59 pliku mainwindow.cpp.

6.7.3.4 on_CameraOrientationLineEdit_textEdited

```
void MainWindow::on_CameraOrientationLineEdit_textEdited (
    const QString & arg1 ) [private], [slot]
```

Definicja w linii 48 pliku mainwindow.cpp.

6.7.3.5 on_CameraOrientationSlider_valueChanged

```
void MainWindow::on_CameraOrientationSlider_valueChanged (
    int value ) [private], [slot]
```

Definicja w linii 43 pliku mainwindow.cpp.

6.7.3.6 on_GloveZoomLineEdit_editingFinished

```
void MainWindow::on_GloveZoomLineEdit_editingFinished ( ) [private], [slot]
```

Definicja w linii 81 pliku mainwindow.cpp.

6.7.3.7 on_GloveZoomLineEdit_textEdited

```
void MainWindow::on_GloveZoomLineEdit_textEdited (
    const QString & arg1 ) [private], [slot]
```

Definicja w linii 70 pliku mainwindow.cpp.

6.7.3.8 on_GloveZoomSlider_valueChanged

```
void MainWindow::on_GloveZoomSlider_valueChanged (
    int value ) [private], [slot]
```

Definicja w linii 65 pliku mainwindow.cpp.

6.7.3.9 on_StartStopButton_clicked

```
void MainWindow::on_StartStopButton_clicked ( ) [private], [slot]
```

Definicja w linii 28 pliku mainwindow.cpp.

6.7.4 Dokumentacja atrybutów składowych

6.7.4.1 InputData

```
Input* MainWindow::InputData [private]
```

Definicja w linii 57 pliku mainwindow.h.

6.7.4.2 ui

```
Ui::MainWindow* MainWindow::ui [private]
```

Definicja w linii 56 pliku mainwindow.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

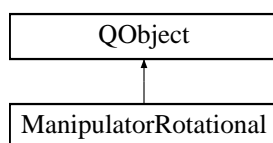
- [mainwindow.h](#)
- [mainwindow.cpp](#)

6.8 Dokumentacja klasy ManipulatorRotational

Definiuje pojęcie manipulatora rotacyjnego.

```
#include <manipulatorrotational.h>
```

Diagram dziedziczenia dla ManipulatorRotational



Metody publiczne

- [ManipulatorRotational](#) (QVector< [Joint](#) *> p_joints, QVector3D p_position=QVector3D(), float p_↵ rotation=float(), Qt3DCore::QEntity *p_rootEntity=nullptr)
- [ManipulatorRotational](#) (const [ManipulatorRotational](#) &p_manipulator)
- void [TransformJointAngles](#) (QVector< float > p_jointAngles)
- [Joint](#) * [getLastJoint](#) ()

Atrybuty chronione

- Qt3DCore::QEntity * [m_rootEntity](#)
- QVector< [Joint](#) * > [m_joints](#)
- QVector3D [m_position](#)
- float [m_rotation](#)

6.8.1 Opis szczegółowy

Inicjuje pozycję stawów w przestrzeni trójwymiarowej

Definicja w linii 22 pliku manipulatorrotational.h.

6.8.2 Dokumentacja konstruktora i destruktora

6.8.2.1 ManipulatorRotational() [1/2]

```
ManipulatorRotational::ManipulatorRotational (
    QVector< Joint *> p_joints,
    QVector3D p_position = QVector3D(),
    float p_rotation = float(),
    Qt3DCore::QEntity * p_rootEntity = nullptr )
```

Definicja w linii 4 pliku manipulatorrotational.cpp.

6.8.2.2 ManipulatorRotational() [2/2]

```
ManipulatorRotational::ManipulatorRotational (
    const ManipulatorRotational & p_manipulator )
```

Definicja w linii 9 pliku manipulatorrotational.cpp.

6.8.3 Dokumentacja funkcji składowych

6.8.3.1 getLastJoint()

```
Joint * ManipulatorRotational::getLastJoint ( )
```

Definicja w linii 26 pliku manipulatorrotational.cpp.

6.8.3.2 TransformJointAngles()

```
void ManipulatorRotational::TransformJointAngles (
    QVector< float > p_jointAngles )
```

Definicja w linii 17 pliku manipulatorrotational.cpp.

6.8.4 Dokumentacja atrybutów składowych

6.8.4.1 m_joints

```
QVector<Joint*> ManipulatorRotational::m_joints [protected]
```

Definicja w linii 27 pliku manipulatorrotational.h.

6.8.4.2 m_position

```
QVector3D ManipulatorRotational::m_position [protected]
```

Definicja w linii 28 pliku manipulatorrotational.h.

6.8.4.3 m_rootEntity

```
Qt3DCore::QEntity* ManipulatorRotational::m_rootEntity [protected]
```

Definicja w linii 26 pliku manipulatorrotational.h.

6.8.4.4 m_rotation

```
float ManipulatorRotational::m_rotation [protected]
```

Definicja w linii 29 pliku manipulatorrotational.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

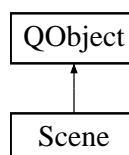
- [manipulatorrotational.h](#)
- [manipulatorrotational.cpp](#)

6.9 Dokumentacja klasy Scene

Modeluje pojecie sceny z elementami 3D.

```
#include <scene.h>
```

Diagram dziedziczenia dla Scene



Metody publiczne

- [Scene](#) (Qt3DCore::QEntity *rootEntity, [MainWindow](#) &p_mainWindow)
- [~Scene](#) ()
- void [SetHandTransformation](#) (QVector< QVector< float >> p_FingerAngles)
- void [SetHandFingertipValues](#) (QVector< int > p_NewValues)

Atrybuty prywatne

- Qt3DCore::QEntity * [m_rootEntity](#)
- [Hand](#) [m_Hand3DModel](#)
- Qt3DExtras::Qt3DWindow * [view](#)
- QWidget * [container](#)
- QSize [screenSize](#)
- QWidget * [widget](#)
- QHBoxLayout * [hLayout](#)
- QVBoxLayout * [vLayout](#)
- Qt3DInput::QInputAspect * [input](#)
- Qt3DRender::QCamera * [cameraEntity](#)
- Qt3DExtras::QFirstPersonCameraController * [camController](#)
- Qt3DCore::QEntity * [lightEntity](#)
- Qt3DRender::QPointLight * [light](#)
- Qt3DCore::QTransform * [lightTransform](#)
- Qt3DCore::QEntity * [lightEntity2](#)
- Qt3DRender::QPointLight * [light2](#)
- Qt3DCore::QTransform * [lightTransform2](#)

6.9.1 Opis szczegółowy

Inicjuje trójwymiatowy model ręki zdefiniowany klasą [Hand](#), za pomocą biblioteki Qt3D, a także inicjalizuje widok, widget, rozmiar okna, układ, wejście danych, kontroler perspektywy i oświetlenie

Definicja w linii 30 pliku scene.h.

6.9.2 Dokumentacja konstruktora i destruktora

6.9.2.1 Scene()

```
Scene::Scene (
    Qt3DCore::QEntity * rootEntity,
    MainWindow & p_mainWindow ) [explicit]
```

Definicja w linii 3 pliku scene.cpp.

6.9.2.2 ~Scene()

```
Scene::~~Scene ( )
```

Definicja w linii 59 pliku scene.cpp.

6.9.3 Dokumentacja funkcji składowych

6.9.3.1 SetHandFingertipValues()

```
void Scene::SetHandFingertipValues (
    QVector< int > p_NewValues )
```

Definicja w linii 69 pliku scene.cpp.

6.9.3.2 SetHandTransformation()

```
void Scene::SetHandTransformation (
    QVector< QVector< float >> p_FingerAngles )
```

Definicja w linii 64 pliku scene.cpp.

6.9.4 Dokumentacja atrybutów składowych

6.9.4.1 camController

```
Qt3DExtras::QFirstPersonCameraController* Scene::camController [private]
```

Definicja w linii 45 pliku scene.h.

6.9.4.2 cameraEntity

```
Qt3DRender::QCamera* Scene::cameraEntity [private]
```

Definicja w linii 44 pliku scene.h.

6.9.4.3 container

```
QWidget* Scene::container [private]
```

Definicja w linii 38 pliku scene.h.

6.9.4.4 hLayout

```
QHBoxLayout* Scene::hLayout [private]
```

Definicja w linii 41 pliku scene.h.

6.9.4.5 input

```
Qt3DInput::QInputAspect* Scene::input [private]
```

Definicja w linii 43 pliku scene.h.

6.9.4.6 light

```
Qt3DRender::QPointLight* Scene::light [private]
```

Definicja w linii 47 pliku scene.h.

6.9.4.7 light2

```
Qt3DRender::QPointLight* Scene::light2 [private]
```

Definicja w linii 50 pliku scene.h.

6.9.4.8 lightEntity

```
Qt3DCore::QEntity* Scene::lightEntity [private]
```

Definicja w linii 46 pliku scene.h.

6.9.4.9 lightEntity2

```
Qt3DCore::QEntity* Scene::lightEntity2 [private]
```

Definicja w linii 49 pliku scene.h.

6.9.4.10 lightTransform

```
Qt3DCore::QTransform* Scene::lightTransform [private]
```

Definicja w linii 48 pliku scene.h.

6.9.4.11 lightTransform2

```
Qt3DCore::QTransform* Scene::lightTransform2 [private]
```

Definicja w linii 51 pliku scene.h.

6.9.4.12 m_Hand3DModel

```
Hand Scene::m_Hand3DModel [private]
```

Definicja w linii 35 pliku scene.h.

6.9.4.13 m_rootEntity

```
Qt3DCore::QEntity* Scene::m_rootEntity [private]
```

Definicja w linii 34 pliku scene.h.

6.9.4.14 screenSize

```
QSize Scene::screenSize [private]
```

Definicja w linii 39 pliku scene.h.

6.9.4.15 view

```
Qt3DExtras::Qt3DWindow* Scene::view [private]
```

Definicja w linii 37 pliku scene.h.

6.9.4.16 vLayout

```
QVBoxLayout* Scene::vLayout [private]
```

Definicja w linii 42 pliku scene.h.

6.9.4.17 widget

```
QWidget* Scene::widget [private]
```

Definicja w linii 40 pliku scene.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [scene.h](#)
- [scene.cpp](#)

Rozdział 7

Dokumentacja plików

7.1 Dokumentacja pliku array.cpp

```
#include "array.h"
```

7.2 Dokumentacja pliku errorhandler.cpp

```
#include "includes.h"
```

7.3 Dokumentacja pliku errorhandler.h

Definicja klasy [ErrorHandler](#).

Komponenty

- class [ErrorHandler](#)
Klasa obsługująca błędy i wyjątki.

7.3.1 Opis szczegółowy

Plik zawiera definicję klasy [ErrorHandler](#)

7.4 Dokumentacja pliku finger.cpp

```
#include "includes.h"
```

7.5 Dokumentacja pliku finger.h

Definicja klasy kwadrat.

```
#include <QObject>
#include <Qt3DCore/QEntity>
#include <QMatrix4x4>
#include <QColor>
#include "joint.h"
#include "manipulatorrotational.h"
#include "fingertip.h"
```

Komponenty

- class [Finger](#)
Modeluje pojęcie palca w przestrzeni trójwymiarowej.

Definicje

- #define [FINGER_JOINTS_COUNT](#) 4
- #define [FINGER_RADIUS](#) 1
- #define [THUMB_JOINT_COUNT](#) 2
- #define [THUMB_JOINT0_LENGTH](#) 7.0f
- #define [THUMB_JOINT1_LENGTH](#) 3.0f
- #define [THUMB_JOINT2_LENGTH](#) 3.0f
- #define [INDEX_JOINT_COUNT](#) 3
- #define [INDEX_FINGER_JOINT0_LENGTH](#) 8.5f
- #define [INDEX_FINGER_JOINT1_LENGTH](#) 5.0f
- #define [INDEX_FINGER_JOINT2_LENGTH](#) 2.5f
- #define [INDEX_FINGER_JOINT3_LENGTH](#) 2.0f
- #define [MIDDLE_JOINT_COUNT](#) 3
- #define [MIDDLE_FINGER_JOINT0_LENGTH](#) 8.0f
- #define [MIDDLE_FINGER_JOINT1_LENGTH](#) 5.3f
- #define [MIDDLE_FINGER_JOINT2_LENGTH](#) 2.8f
- #define [MIDDLE_FINGER_JOINT3_LENGTH](#) 2.3f
- #define [RING_JOINT_COUNT](#) 3
- #define [RING_FINGER_JOINT0_LENGTH](#) 7.5f
- #define [RING_FINGER_JOINT1_LENGTH](#) 4.5f
- #define [RING_FINGER_JOINT2_LENGTH](#) 2.8f
- #define [RING_FINGER_JOINT3_LENGTH](#) 2.4f
- #define [PINKY_JOINT_COUNT](#) 3
- #define [PINKY_JOINT0_LENGTH](#) 6.3f
- #define [PINKY_JOINT1_LENGTH](#) 3.3f
- #define [PINKY_JOINT2_LENGTH](#) 2.0f
- #define [PINKY_JOINT3_LENGTH](#) 2.3f

7.5.1 Opis szczegółowy

Plik zawiera definicję klasy finger, która jest klasą pochodną klasy QObject

7.5.2 Dokumentacja definicji

7.5.2.1 FINGER_JOINTS_COUNT

```
#define FINGER_JOINTS_COUNT 4
```

Definicja w linii 12 pliku finger.h.

7.5.2.2 FINGER_RADIUS

```
#define FINGER_RADIUS 1
```

Definicja w linii 13 pliku finger.h.

7.5.2.3 INDEX_FINGER_JOINT0_LENGTH

```
#define INDEX_FINGER_JOINT0_LENGTH 8.5f
```

Definicja w linii 21 pliku finger.h.

7.5.2.4 INDEX_FINGER_JOINT1_LENGTH

```
#define INDEX_FINGER_JOINT1_LENGTH 5.0f
```

Definicja w linii 22 pliku finger.h.

7.5.2.5 INDEX_FINGER_JOINT2_LENGTH

```
#define INDEX_FINGER_JOINT2_LENGTH 2.5f
```

Definicja w linii 23 pliku finger.h.

7.5.2.6 INDEX_FINGER_JOINT3_LENGTH

```
#define INDEX_FINGER_JOINT3_LENGTH 2.0f
```

Definicja w linii 24 pliku finger.h.

7.5.2.7 INDEX_JOINT_COUNT

```
#define INDEX_JOINT_COUNT 3
```

Definicja w linii 20 pliku finger.h.

7.5.2.8 MIDDLE_FINGER_JOINT0_LENGTH

```
#define MIDDLE_FINGER_JOINT0_LENGTH 8.0f
```

Definicja w linii 27 pliku finger.h.

7.5.2.9 MIDDLE_FINGER_JOINT1_LENGTH

```
#define MIDDLE_FINGER_JOINT1_LENGTH 5.3f
```

Definicja w linii 28 pliku finger.h.

7.5.2.10 MIDDLE_FINGER_JOINT2_LENGTH

```
#define MIDDLE_FINGER_JOINT2_LENGTH 2.8f
```

Definicja w linii 29 pliku finger.h.

7.5.2.11 MIDDLE_FINGER_JOINT3_LENGTH

```
#define MIDDLE_FINGER_JOINT3_LENGTH 2.3f
```

Definicja w linii 30 pliku finger.h.

7.5.2.12 MIDDLE_JOINT_COUNT

```
#define MIDDLE_JOINT_COUNT 3
```

Definicja w linii 26 pliku finger.h.

7.5.2.13 PINKY_JOINT0_LENGTH

```
#define PINKY_JOINT0_LENGTH 6.3f
```

Definicja w linii 39 pliku finger.h.

7.5.2.14 PINKY_JOINT1_LENGTH

```
#define PINKY_JOINT1_LENGTH 3.3f
```

Definicja w linii 40 pliku finger.h.

7.5.2.15 PINKY_JOINT2_LENGTH

```
#define PINKY_JOINT2_LENGTH 2.0f
```

Definicja w linii 41 pliku finger.h.

7.5.2.16 PINKY_JOINT3_LENGTH

```
#define PINKY_JOINT3_LENGTH 2.3f
```

Definicja w linii 42 pliku finger.h.

7.5.2.17 PINKY_JOINT_COUNT

```
#define PINKY_JOINT_COUNT 3
```

Definicja w linii 38 pliku finger.h.

7.5.2.18 RING_FINGER_JOINT0_LENGTH

```
#define RING_FINGER_JOINT0_LENGTH 7.5f
```

Definicja w linii 33 pliku finger.h.

7.5.2.19 RING_FINGER_JOINT1_LENGTH

```
#define RING_FINGER_JOINT1_LENGTH 4.5f
```

Definicja w linii 34 pliku finger.h.

7.5.2.20 RING_FINGER_JOINT2_LENGTH

```
#define RING_FINGER_JOINT2_LENGTH 2.8f
```

Definicja w linii 35 pliku finger.h.

7.5.2.21 RING_FINGER_JOINT3_LENGTH

```
#define RING_FINGER_JOINT3_LENGTH 2.4f
```

Definicja w linii 36 pliku finger.h.

7.5.2.22 RING_JOINT_COUNT

```
#define RING_JOINT_COUNT 3
```

Definicja w linii 32 pliku finger.h.

7.5.2.23 THUMB_JOINT0_LENGTH

```
#define THUMB_JOINT0_LENGTH 7.0f
```

Definicja w linii 16 pliku finger.h.

7.5.2.24 THUMB_JOINT1_LENGTH

```
#define THUMB_JOINT1_LENGTH 3.0f
```

Definicja w linii 17 pliku finger.h.

7.5.2.25 THUMB_JOINT2_LENGTH

```
#define THUMB_JOINT2_LENGTH 3.0f
```

Definicja w linii 18 pliku finger.h.

7.5.2.26 THUMB_JOINT_COUNT

```
#define THUMB_JOINT_COUNT 2
```

Definicja w linii 15 pliku finger.h.

7.6 Dokumentacja pliku fingertip.cpp

```
#include "includes.h"
```

7.7 Dokumentacja pliku fingertip.h

Definicja klasy [Fingertip](#).

```
#include <QObject>
#include <Qt3DCore/QEntity>
#include <Qt3DCore/QTransform>
#include <Qt3DExtras/QPhongMaterial>
#include <QMatrix4x4>
#include "joint.h"
```

Komponenty

- class [Fingertip](#)

Opisuje opuszki palców.

Definicje

- `#define SPHERE_RADIUS 1.5`
- `#define SPHERE_RINGS_COUNT 20`
- `#define SPHERE_SLICES_COUNT 20`
- `#define SPHERE_SCALE 0.05`

7.7.1 Opis szczegółowy

Plik zawiera definicję klasy [Fingertip](#), która jest klasą pochodną klasy `QObject`

7.7.2 Dokumentacja definicji

7.7.2.1 SPHERE_RADIUS

```
#define SPHERE_RADIUS 1.5
```

Definicja w linii 20 pliku `fingertip.h`.

7.7.2.2 SPHERE_RINGS_COUNT

```
#define SPHERE_RINGS_COUNT 20
```

Definicja w linii 21 pliku `fingertip.h`.

7.7.2.3 SPHERE_SCALE

```
#define SPHERE_SCALE 0.05
```

Definicja w linii 23 pliku `fingertip.h`.

7.7.2.4 SPHERE_SLICES_COUNT

```
#define SPHERE_SLICES_COUNT 20
```

Definicja w linii 22 pliku `fingertip.h`.

7.8 Dokumentacja pliku hand.cpp

```
#include "includes.h"
```

7.9 Dokumentacja pliku hand.h

Definicja klasy [Hand](#).

```
#include <QObject>
#include <Qt3DCore/QEntity>
#include "joint.h"
#include "manipulatorrotational.h"
#include "fingertip.h"
#include "finger.h"
```

Komponenty

- class [Hand](#)

Modeluje pojęcie dłoni.

Definicje

- #define [FINGER_COUNT](#) 5
- #define [THUMB_FINGER_OFFSET](#) 2.0
- #define [INDEX_FINGER_OFFSET](#) 1.0
- #define [MIDDLE_FINGER_OFFSET](#) 0.0
- #define [RING_FINGER_OFFSET](#) -1.0
- #define [PINKY_FINGER_OFFSET](#) -2.0
- #define [THUMB_FINGER_ROTATION](#) -30
- #define [INDEX_FINGER_ROTATION](#) -10
- #define [MIDDLE_FINGER_ROTATION](#) 0.0
- #define [RING_FINGER_ROTATION](#) 10
- #define [PINKY_FINGER_ROTATION](#) 20

7.9.1 Opis szczegółowy

Plik zawiera definicję klasy [Hand](#), która składa się z elementów klasy [Finger](#)

7.9.2 Dokumentacja definicji

7.9.2.1 FINGER_COUNT

```
#define FINGER_COUNT 5
```

Definicja w linii 11 pliku hand.h.

7.9.2.2 INDEX_FINGER_OFFSET

```
#define INDEX_FINGER_OFFSET 1.0
```

Definicja w linii 14 pliku hand.h.

7.9.2.3 INDEX_FINGER_ROTATION

```
#define INDEX_FINGER_ROTATION -10
```

Definicja w linii 20 pliku hand.h.

7.9.2.4 MIDDLE_FINGER_OFFSET

```
#define MIDDLE_FINGER_OFFSET 0.0
```

Definicja w linii 15 pliku hand.h.

7.9.2.5 MIDDLE_FINGER_ROTATION

```
#define MIDDLE_FINGER_ROTATION 0.0
```

Definicja w linii 21 pliku hand.h.

7.9.2.6 PINKY_FINGER_OFFSET

```
#define PINKY_FINGER_OFFSET -2.0
```

Definicja w linii 17 pliku hand.h.

7.9.2.7 PINKY_FINGER_ROTATION

```
#define PINKY_FINGER_ROTATION 20
```

Definicja w linii 23 pliku hand.h.

7.9.2.8 RING_FINGER_OFFSET

```
#define RING_FINGER_OFFSET -1.0
```

Definicja w linii 16 pliku hand.h.

7.9.2.9 RING_FINGER_ROTATION

```
#define RING_FINGER_ROTATION 10
```

Definicja w linii 22 pliku hand.h.

7.9.2.10 THUMB_FINGER_OFFSET

```
#define THUMB_FINGER_OFFSET 2.0
```

Definicja w linii 13 pliku hand.h.

7.9.2.11 THUMB_FINGER_ROTATION

```
#define THUMB_FINGER_ROTATION -30
```

Definicja w linii 19 pliku hand.h.

7.10 Dokumentacja pliku includes.h

Dołączenie bibliotek.

```
#include <iostream>
#include <cmath>
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QtWidgets/QApplication>
#include <QtWidgets/QWidget>
#include <QtWidgets/QHBoxLayout>
#include <QtGui/QScreen>
#include <QObject>
#include <QVector3D>
#include <QMatrix3x3>
#include <QMatrix4x4>
#include <QQuaternion>
#include <QtGlobal>
#include <QDataStream>
#include <QtBluetooth>
#include <QtSerialPort>
#include <QGuiApplication>
#include <Qt3DCore/qentity.h>
#include <Qt3DRender/qcamera.h>
#include <Qt3DRender/qcameralens.h>
#include <Qt3DRender/qmesh.h>
#include <Qt3DRender/qtechnique.h>
#include <Qt3DRender/qmaterial.h>
#include <Qt3DRender/qeffect.h>
#include <Qt3DRender/qtexture.h>
#include <Qt3DRender/qrenderpass.h>
#include <Qt3DRender/qsceneloader.h>
#include <Qt3DRender/qpointlight.h>
#include <Qt3DRender/qrenderaspect.h>
#include <Qt3DCore/qtransform.h>
#include <Qt3DCore/qaspectengine.h>
#include <Qt3DInput/qinputaspect.h>
#include <Qt3DExtras/qforwardrender.h>
#include <Qt3DExtras/qt3dwindow.h>
#include <Qt3DExtras/qfirstpersoncameracontroller.h>
#include <Qt3DExtras/QPhongMaterial>
#include <Qt3DExtras/QCylinderMesh>
#include <Qt3DExtras/QSphereMesh>
#include "errorhandler.h"
#include "joint.h"
#include "manipulatorrotational.h"
#include "fingertip.h"
#include "finger.h"
#include "hand.h"
#include "scene.h"
#include "input.h"
```

7.10.1 Opis szczegółowy

Plik zawiera biblioteki dołączone do projektu

7.11 Dokumentacja pliku input.cpp

```
#include "includes.h"
```

7.12 Dokumentacja pliku input.h

Definicja klasy [Input](#).

```
#include <QtBluetooth>
#include <QtSerialPort>
```

Komponenty

- class [Input](#)

Klasa jest narzędziem do pobierania danych.

7.12.1 Opis szczegółowy

Plik zawiera definicję klasy [Input](#), jest klasą pochodną klas [QBluetoothTransferManager](#) oraz [QSerialPort](#)

7.13 Dokumentacja pliku joint.cpp

```
#include "includes.h"
```

7.14 Dokumentacja pliku joint.h

Definicja klasy [Joint](#).

```
#include <QObject>
#include <Qt3DCore/QEntity>
#include <Qt3DCore/QTransform>
#include <QMatrix4x4>
#include <QColor>
```

Komponenty

- class [Joint](#)

Modeluje pojęcie stawu.

Definicje

- `#define CYLINDER_RINGS_COUNT 100`
- `#define CYLINDER_SLICES_COUNT 20`

7.14.1 Opis szczegółowy

Plik zawiera definicję klasy [Joint](#), która jest klasą pochodną klasy `QObject`

7.14.2 Dokumentacja definicji

7.14.2.1 CYLINDER_RINGS_COUNT

```
#define CYLINDER_RINGS_COUNT 100
```

Definicja w linii 12 pliku `joint.h`.

7.14.2.2 CYLINDER_SLICES_COUNT

```
#define CYLINDER_SLICES_COUNT 20
```

Definicja w linii 13 pliku `joint.h`.

7.15 Dokumentacja pliku `main.cpp`

```
#include "includes.h"
```

Funkcje

- `int main (int argc, char *argv[])`

7.15.1 Dokumentacja funkcji

7.15.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definicja w linii 3 pliku main.cpp.

7.16 Dokumentacja pliku mainwindow.cpp

```
#include "includes.h"
```

7.17 Dokumentacja pliku mainwindow.h

Definicja klasy [MainWindow](#).

```
#include <QMainWindow>
#include "input.h"
```

Komponenty

- class [MainWindow](#)
Modeluje strukturę GUI programu.

Przestrzenie nazw

- [Ui](#)

7.17.1 Opis szczegółowy

Plik zawiera definicję klasy [MainWindow](#), która jest klasą pochodną klasy QMainWindow

7.18 Dokumentacja pliku manipulatorrotational.cpp

```
#include "includes.h"
```

7.19 Dokumentacja pliku manipulatorrotational.h

Definicja klasy [ManipulatorRotational](#).

```
#include <QObject>
#include <Qt3DCore/QEntity>
#include "joint.h"
```

Komponenty

- class [ManipulatorRotational](#)
Definiuje pojęcie manipulatora rotacyjnego.

7.19.1 Opis szczegółowy

Plik zawiera definicję klasy [ManipulatorRotational](#), która jest klasą pochodną klasy `QObject`

7.20 Dokumentacja pliku scene.cpp

```
#include "includes.h"
```

7.21 Dokumentacja pliku scene.h

Definicja klasy [Scene](#).

```
#include <QObject>
#include <Qt3DCore>
#include <Qt3DExtras>
#include <Qt3DRender>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include "mainwindow.h"
#include "joint.h"
#include "finger.h"
#include "hand.h"
```

Komponenty

- class [Scene](#)
Modeluje pojęcie sceny z elementami 3D.

7.21.1 Opis szczegółowy

Plik zawiera definicję klasy [Scene](#), która jest klasą pochodną klasy `QObject`