|  |  |
| --- | --- |
| Project | **Specification of Sensor Interface for Cyber and Physical World**  <<https://sagroups.ieee.org/2888.1/> **>** |
| Title | **Syntax and semantics of large space VR training system input devices capabilities** |
| DCN | **2888-22-0032-00-0001** |
| Date Submitted | **Feb. 12th, 2022** |
| Source(s) | Sang-Kyun Kim, [goldmunt@gmail.com](mailto:goldmunt@gmail.com) (Myongji University)  Min Hyuk Jeong, [jmh8900@gmail.com](mailto:jmh8900@gmail.com) (Myongji University) |
| Re: |  |
| Abstract | This contribution illustrates the basic JSON schema structure for representing input device capabilities of a large space VR training system in a standardized data format. The semantics and examples of input device capabilities are presented. |
| Purpose | To start discussion on purpose of the standard |
| Notice | This document has been prepared to assist the IEEE 2888 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 2888 may make this contribution public. |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> |

# Introduction

This contribution illustrates the basic JSON schema structure for representing input device capabilities of a large space VR training system in a standardized data format. The semantics and examples of input device capabilities are presented.

# Button sensor capability

## General

This sub-clause specifies a capability of a button sensor.

## Syntax

|  |
| --- |
| "buttonSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedNumberOfButtons": {  "type": "integer",  }  }  } |

## Semantics

Semantics of the buttonSensorCapabilityData:

| Name | Definition |
| --- | --- |
| buttonSensorCapabilityData | Tool for describing a button sensor capability |
| supportedNumberOfButtons | It describes the number of supported buttons |

## Examples

The botton sensor supports 8 buttons in this example.

|  |
| --- |
| buttonSensorCapabilityData:{  "sensorCapabilityBaseType": {},  "supportedNumberOfButtons": 8  } |

# Analog sensor capability

## General

This sub-clause specifies a capability of an analog sensor.

## Syntax

|  |
| --- |
| "analogSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedNumberOfChannels": {  "type": "integer",  }  }  } |

## Semantics

Semantics of the analogSensorCapabilityData:

| Name | Definition |
| --- | --- |
| analogSensor CapabilityData | Tool for describing an analog sensor capability |
| supportedNumberOfChannels | It describes the number of supported channels |

## Examples

The analog sensor supports 4 channels in this example.

|  |
| --- |
| analaogSensorCapabilityData:{  "sensorCapabilityBaseType": {},  "supportedNumberOfChannels": 4  } |

# Dial sensor capability

## General

This sub-clause specifies a capability of a dial sensor.

## Syntax

|  |
| --- |
| "dialSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedNumberOfDials": {  "type": "integer",  }  }  } |

## Semantics

Semantics of the dialSensorCapabilityData:

| Name | Definition |
| --- | --- |
| dialSensor CapabilityData | Tool for describing a dial sensor capability |
| supportedNumberOfDials | It describes the number of supported dials |

## Examples

The dial sensor supports 4 channels in this example.

|  |
| --- |
| dialSensorCapabilityData:{  "sensorCapabilityBaseType": {  "minValue": 0.0,  "maxValue": 90.0,  "unit": "degree"  },  "supportedNumberOfDials": 4  } |

# Haptic sensor capability

## General

This sub-clause specifies a capability of a haptic sensor.

## Syntax

|  |
| --- |
| "hapticSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedNumberOfSimultaneousTouches": {  "type": "integer",  }  }  } |

## Semantics

Semantics of the hapticSensorCapabilityData:

| Name | Definition |
| --- | --- |
| hapticSensor CapabilityData | Tool for describing a haptic sensor capability |
| supportedNumberOfSimultaneousTouches | It describes the number of supported simultaneous touches |

## Examples

The haptic sensor supports 4 touches simultaneously in this example.

|  |
| --- |
| hapticSensorCapabilityData:{  "sensorCapabilityBaseType": {},  "supportedNumberOfSimultaneousTouches": 4  } |

# Glove sensor capability

## General

This sub-clause specifies a capability of a glove sensor.

## Syntax

|  |
| --- |
| "gloveSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedHandSide": {  "type": "string",  "enum": ["left", "right", "both"]  }  }  } |

## Semantics

Semantics of the gloveSensorCapabilityData:

| Name | Definition |
| --- | --- |
| gloveSensor CapabilityData | Tool for describing a glove sensor capability |
| supportedHandSide | It describes supported hand side |

## Examples

In this example, the glove sensor supports both hand sides and can measure 0 to 60 degrees.

|  |
| --- |
| gloveSensorCapabilityData:{  "sensorCapabilityBaseType": {  "minValue": 0.0  "maxValue": 60.0  "unit": "degree"  },  "supportedHandSide": "both"  } |

# IMU sensor capability

## General

This sub-clause specifies a capability of an IMU sensor.

## Syntax

|  |
| --- |
| "IMUSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  }  } |

## Semantics

Semantics of the IMUSensorCapabilityData:

| Name | Definition |
| --- | --- |
| IMUSensorCapabilityData | Tool for describing an IMU sensor capability |

## Examples

The IMU sensor can measure -180 to 180 degrees in this example.

|  |
| --- |
| IMUSensorCapabilityData:{  "sensorCapabilityBaseType": {  "minValue": -180.0  "maxValue": 180.0  "unit": "degree"  },  } |

# Rigidbody sensor capability

## General

This sub-clause specifies a capability of a rigid body sensor.

## Syntax

|  |
| --- |
| "rigidbodySensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  }  } |

## Semantics

Semantics of the rigidbodySensorCapabilityData:

| Name | Definition |
| --- | --- |
| rigidbodySensor CapabilityData | Tool for describing a rigid body sensor capability |

## Examples

The rigidbody sensor can measure 0 to 360 degrees in this example.

|  |
| --- |
| analaogSensorCapabilityData:{  "sensorCapabilityBaseType": {  "minValue": 0.0,  "maxValue": 360.0  },  } |

# Bend sensor capability

## General

This sub-clause specifies a capability of a bend sensor.

## Syntax

|  |
| --- |
| "bendSensorCapabilityData": {  "type": "object",  "properties": {  "sensorCapabilityBaseType": {  "$ref": "#/definitions/sensorCapabilityBaseType"  },  "supportedNumberOfJoints": {  "type": "integer",  }  "supportedAxis": {  "type": "integer",  }  }  } |

## Semantics

Semantics of the bendSensorCapabilityData:

| Name | Definition |
| --- | --- |
| bendSensorCapabilityData | Tool for describing a bend sensor capability |
| supportedNumberOfJoints | It describes the number of supported joints |
| supportedAxis | It describes the supported axis on each joint. Each joint can have 1 to 3 axis. In the case of two axes, x and y, the case of three axes, x, y, and z, are described in order |

## Examples

In this example, the bend sensor supports 8 joints and measures -180 to 180(degrees). Every joint can measure x, y, and z-direction.

|  |
| --- |
| bendSensorCapabilityData:{  "sensorCapabilityBaseType": {  "minValue": -180.0,  "maxValue": 180.0,  "unit": "degree"  },  "supportedNumberOfJoints": 8,  "supportedAxis": 3  } |

# Conclusions

We recommend accepting the proposed large space VR training system input device capabilities.