

Homogenizing Physical and Virtual Coordinate Spaces

Arnaud Loonstra
Whitepaper, July 15 2015
Stichting z25.org
arnaud@sphaero.org

Abstract

In this paper we argue for a unification of coordinate spaces for use in mixed reality setups. The paper introduces most used coordinate systems and proposes a system for unification based on existing coordinate systems. An example is given at the end of the paper.

1. Introduction

Our environment is nowadays filled with technology. Mobile phones, smart watches, game interfaces, sensors, camera's, rfid, etc. This means all this technology can somehow relate to the physical realm. However there are no standards for mapping a physical realm to a digital/virtual realm. One can create a mapping like one needs. However when the amount of devices grows this can be a challenging task. You hope you chose a sensible and flexible mapping set for every device. Apart from making it work it would be more beneficial if the mapping is 'making sense'. If coordinate systems would not map 1 on 1 then for every change one would need to think about the translation from one coordinate system into the other. This is unintuitive, error prone and making adoption of a clear mental model more difficult.

In this document we outline the results of our research into homogenizing different coordinates systems from the physical and digital realm. We propose a mapping system to homogenize the realm's coordinate systems. The proposed mapping is tested for use in the ZOCP protocol[1].

2. What is a coordinate system

A coordinate system is a numbering system to uniquely determine a position in a space. A simple example of coordinate system is a position on a line. When an origin, position 0, is determined we can relate the position of any unique point on that line to a positive or negative number.

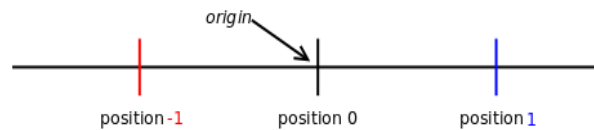


Illustration 1: coordinates on a line

When we want to determine a position on a plane we need to add another line to determine the position using 2 numbers. We call this a 2 dimensional coordinate system.

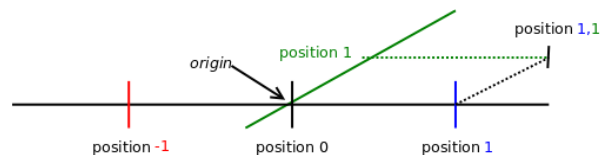


Illustration 2: coordinate on a plane

We can repeat this to position into 3 dimensions. This can continue indefinitely. However in our physical presence we are used to a three dimensional space. Sometimes we add time as a fourth dimensional coordinate. It is important to understand that the limitation of our physical presence to 3 dimensions is not valid for mathematical spaces.

Besides describing linear positions we can also define angular positions which are usually used for describing orientations. Just think about a compass. It defines North, South, East and West directions. It also defines this as numbers North being 0 degrees and South being 180 degrees. Again this can be repeated to multiple orientation dimensions. We will describe this in detail in this paper.

3. Physical coordinate systems

In our daily life we are used to many different coordinate systems. When we drive our car to work we are used to kilometers in most countries but in others miles might be used. When sail a ship we will probably use nautical miles and perhaps when we are flying to another planet we are talking about light years. All these coordinate systems are essentially the same. They only differ in the units they use. We will

describe the most used coordinate systems and reason why we chose to use them.

3.1. Metric System

The Metric System[2] is a system introduced by the French in 1799. Nowadays it is also used under the synonym “SI” or “International System of Units”. It is officially the most used measuring system in almost all countries. This is also the most profound reason why we chose to adopt this system as a standard for mapping physical and virtual realms. The only countries that do not use the metric system are America, Liberia and Myanmar(Burma).

The foundations of the metric system are the 'meter' and the 'kilogram'. The metric system was designed to be universal and based on natural phenomena. For example one meter was initially defined to be one ten millionth of the distance from the equator to the North Pole through Paris. A kilogram was defined as the mass of 1/1000th of a cubic meter of pure water at its melting point. Future developments showed more accurate ways to determine a meter's length and by 1983 a meter was defined by the constant of the speed of light.

It is important to realize that these metric units are based on natural reproducible facts. Many other units are based on the metric system and therefore show coherence. We will show later that this is important for the digital realm as well. Since the metric system enjoys broad adoption we will adopt this system as a foundation.

In 1999 the \$125 million dollar Mars Climate Orbiter crashed into Mars because the American spacecraft engineers calculated using American units while the thrust engineers were expecting units specified in the metric system. This illustrates the importance of having agreed standards hence this research.

3.2. Orientation

The most common orientation system is the compass bearing. It was invented somewhere around 200BC and we are still using the North, South, East, West orientations today. The compass bearing describes a single dimensional orientation using degrees as an angular unit. It is unknown how the degree unit originated and what it is based on. The degree is not an SI unit. It is however mentioned as a 'accepted' unit. The official unit in SI is a 'radian'. The radian is a more practical unit since it is easier to use for mathematical operations. The radian is defined as being the length of half a circle with a radius of 1. So if you take half a circle with a one meter radius the length of this half circle will be 1 radian which is 3.1415926535... meter. This is the famous Pi (π)

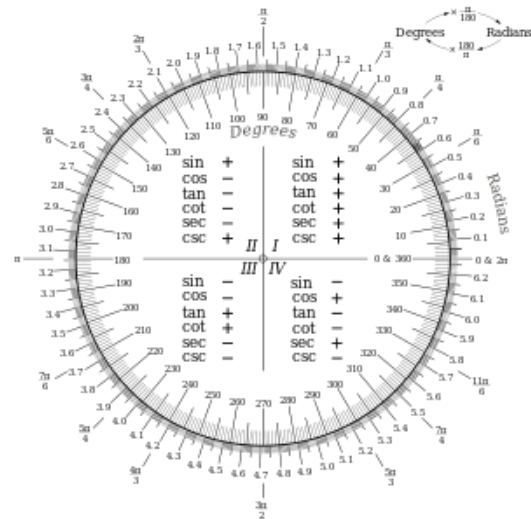


Illustration 3: A chart to convert between degrees and radians (courtesy of Wikipedia)

number. Illustration 3 is a chart consisting of different angular coordinates and show how to convert between them.

Ever since we are looking up to the stars we know that a compass bearing orientation is not enough to describe our orientation. There are at least three units needed to describe our orientation. These are often called yaw, pitch and roll and stem from Euler Angles. Other system might refer to these as heading, pitch and bank. As both define pitch this can be the source

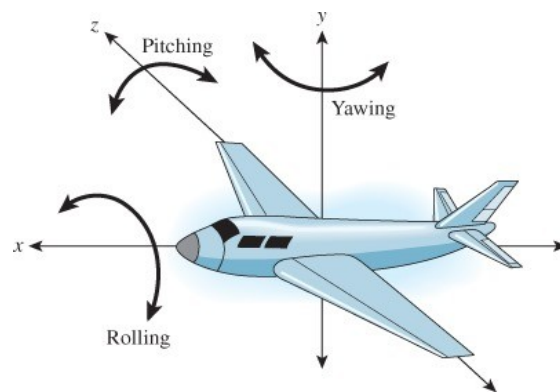


Illustration 4: Euler Angles's Pitch, Yaw and Roll (Courtesy of gamesfromscratch.com)

of confusion hence we discard this notation.

An Euler Angle has its limitations. Imagine turning to an orientation with the maximum values. Now try turning beyond that orientation. This called the 'Gimbal Lock'. Because of the 'Gimbal Lock' quaternion orientation might be suited better. As Quaternions are very mathematical we only mention their existence as a coordinate orientation and will not elaborate further[3].

3.3. Time

Time is one of the seven fundamentals in the metric system (SI). Time is used to describe the order of events from past to present to future. It is thus a one dimensional coordinate system. A time unit in the metric system is defined as a second. Initially time was determined by the cycle of the moon and the sun but nowadays a second is defined in terms of radiation emitted by caesium atoms.

3.4. Positioning on earth

A coordinate system known by many because of its adoption by navigations devices using the Global Positioning System (GPS) is called WGS84[4]. It was established in 1984 and its latest revision is of 2004. WGS84 is a coordinate system to describe any location on earth with a 2cm accuracy. It is important to understand that WGS84 is a 2D coordinate system on a mathematical sphere. This means it can never describe the height of your position. As WGS84, through the GPS system, has broad adoption it would be unwise to not adopt it. However it is wise to look further as one could imagine doing something outside of earths boundaries (e.g. satellites, space probes). Some alternatives to WGS84 are the European Terrestrial Reference System 1989 and North American Datum 1983 however these have the same shortcoming as WGS84. The International Astronomical Union (IAU) has adopted the International Celestial Reference System (ICRS). ICRS is a high precision positional coordinate system. It has its origin (position 0,0,0) at the solar system barycenter. This would be an ideal reference coordinate system. On the other hand to use it practically will be very hard because of the simple fact that our planet earth is flying around the sun at an incredible speed. Any ICRS coordinate on earth would be subject to a constant change and a fixed position depends on time. So to remain practical we would adopt the common coordinate system used by GPS receivers, but to remain ambitious we would adopt the ICRS coordinate system

4. Virtual coordinate systems

The virtual or digital realm is essentially an abstraction. Inside the computer there is no such thing as a virtual reality that can be regarded as the physical reality. The virtual reality can computationally simulate a physical reality in 3 dimensions but since it is an abstraction the computer can continue beyond three dimensions. However for our purposes of mapping a physical reality to a virtual reality three dimension will do fine. We will look into coordinate systems similar to those in the physical realm used in

the digital realms. As most of our digital realities are related to OpenGL it might be sensible to follow its approach as close as possible.

4.1. OpenGL

The most used functionality of OpenGL is to translate a three dimensional scene into a two dimensional representation. This is basically what happens when you are playing a 3D game on your monitor. OpenGL defines several coordinate spaces to support converting a 3d space into a 2d representation. However OpenGL, being a general purpose language, does not dictate any standard into mapping a physical coordinate space to a virtual. It therefore provides any desired way hence our need to settle on one approach to prevent dealing with different interpretations.

OpenGL defines "Object Space". Three-dimensional objects are defined a coordinate space convenient for describing the object's properties (vertices, normals etc). Or to put in layman's terms; in case of a pyramid the 5 points (4 bottom, 1 top) of the

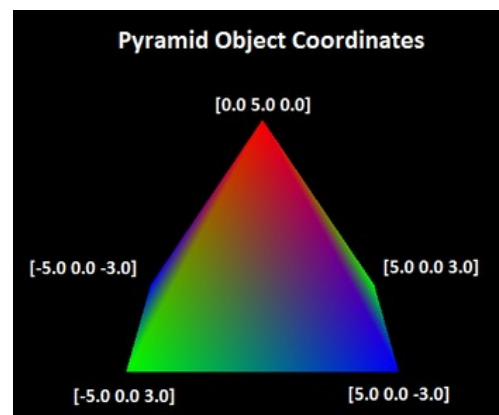


Illustration 5: Object Space coordinates

pyramid are described in "Object Space" coordinates.

To see this pyramid in a context of a scene OpenGL defines "World Space". The pyramid's location is described in "World Space" coordinates.

In order for the pyramid's position to be described in this space it needs an 'origin' point. The origin point is described in the object's "Object Space". To understand this from a physical perspective imagine you are holding the pyramid in your hand. The exact position where you are holding the pyramid is the pyramid's origin.

OpenGL has a notion of 'units'. However, if an object has a size of 3 units this could mean anything. Is a unit a meter, a millimeter, a lightyear? To OpenGL this is unimportant however if one wants to relate to our physical space in an absolute way we need to specify what a 'unit' is. In OpenGL a unit can be different in every coordinate space. Suppose your pyramid has a unit size of 1 centimeter. In OpenGL it

is possible to define “World Space” in meters. In order to have a correct representation of the pyramid we need to transform the “Object Space” coordinate system into the “World Space” coordinate system. In OpenGL this is called “Model Transformation”. In our example we need to scale the “Object Space” coordinate system by a factor of 10 to match it to “World Space” coordinates. In OpenGL the “Model Transformation Matrix” describes this transformation. It is called a “Matrix” since the transformation between coordinate space is done using mathematical matrices. To understand this fully you have to understand linear algebra. However, at least remember that matrix calculation is used to transform from one coordinate space to another.

Finally in case we are displaying this OpenGL setup we need to transform “World Space” into “Eye Space”. This is similar to “Object Space” transformation thus described in the “View Matrix”. To imagine this you need just need to think of the camera filming the scene. The camera has a position, orientation, etc in “World Space” coordinates. To display a scene through the camera all coordinates must be described in coordinate space of the camera.

To map the OpenGL coordinate system to a physical coordinate system we need to take some physical properties into account. We all know that when we hold an apple in the air and let go it will drop. You should know that this is how Newton discovered the basic laws of gravitation. The apple will fall to the floor with an acceleration of 9.81 m/s². Suppose we would want to simulate this in OpenGL. We could write our own methods to simulate dropping an apple but we will probably use a physics engine as we don't want to reinvent the wheel. A physics engine simulates essential properties of the physical realm. It would make sense to make use of the same gravitational laws as we have in the physical realm. Since we will probably remain on earth most of the time it makes sense to use the acceleration value of 9.81 m/s². This implies we are calculating in meters. Therefore it makes sense to set an OpenGL unit to 1 meter as well to guarantee numbers remain the same. It is convenient to have equal number so we don't have to transform numbers for each coordinate system.

Finally we need to define what is the positive direction and what negative. There are actually two main coordinate systems: Right-Handed and Left-Handed. Why those names?

1. Stretch your right arm and form a 90 degree angle with your elbow
2. Point your right hand thumb to the right side
3. Point your right hand finger up
4. Point your right hand middle finger in your direction

This is the Right-handed coordinate system. Your fingers are now pointing in the positive direction.

OpenGL uses the Right-Handed system. Obviously if you would do this with your left hand you would get the Left-Handed system.

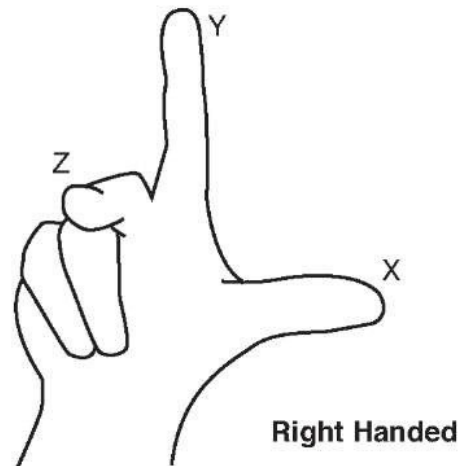


Illustration 6: Right hand fingers are pointing in the positive direction

Actually we haven't set this for our physical coordinate system either. To match directions in the physical realm we have to face north. By this the RHS will be valid for the physical realm as well as this matches WGS84 directions.

We now have described the coordinate systems of OpenGL we need. We have “World Space” coordinates to define the location of every object in our world. We have “Object Space” coordinates to define the dimensional properties of an individual object. Why two sets of coordinates? Imagine your newly bought DIY construction cupboard. If you are constructing the cupboard you are not thinking of how it is constructed in dimensional terms of your living room. The manufacturer also cannot give you instructions based on the dimensions of your living room because of the simple fact that the manufacturer doesn't know the dimensions of your living room.

4.2. Digital Time

Time in a computer is a more complicated coordinate than in the physical realm. If you switch a computer off it has no notion of time. If you power it on it cannot know what time it is. For the computer time is a very relative coordinate. The only way it can determine a time unit is by its cpu clock. However this is not the most stable reference and often causes drift. Therefore most computers are equipped with an on-board clock. This clock ticks even if your computer is off. Even then time is unreliable as it must be set manually. Networked computers are able synchronize their clocks using a protocol like the Network Time

Protocol (NTP)[5]. This protocol ensures the computers have their clocks synchronized with maximally a few milliseconds drift. However without an external source time can still not be reliably mapped onto physical time. On internet a hierarchy of NTP servers provide a reliable source to synchronize clocks with. The GPS system (WGS84) also provides a very reliable time signal with an accuracy of +/-10 nanoseconds.

5. Mapping physical coordinates to virtual

We believe that having a sensible coordinate systems improves workflow when working in both realms. As said before transforming coordinates systems is an error prone activity as was illustrated by the crash of the Mars Climate Orbiter. We have experienced situations in which coordinate systems needed translation causing a lot of strain on rapid prototyping developments and situations which demanded improvising. Non-homogenized coordinate systems also proved much harder to debug again straining development progression. If we settle on the coordinate system provided by the metric system and the coordinate systems of OpenGL as described before we can now make a mapping between the physical and virtual realm.

We now describe the used coordinates starting in the physical realm and moving into the virtual realm. We are describing the coordinates as a hierarchy. When you have a chair in a room and you move the room the chair will move along. The relation of the objects is described by this hierarchy.

Our physical setup consists of a room with dimension of 3x4x2 meters (LxWxH) with its origin at GPS coordinates: 52.1153016,5.1389998. We call this our physical root.

```
root:
WGS84: (52.1153016, 5.1389998)
Room:
  position: (0,0,0)
  dimension: (2.0, 4.0, 2.0)
  orientation: (radians? Euler?)
  matrix: (describes, position, rotation and
           scale in 9 floats)
```

Inside the room we put a chair. The chair has a dimension of 60x80x90 (LxWxH) centimeter and is placed in the left corner of the room opposite the origin of the room.

```
Chair:
  position: (1,5, 3.5, 0)
  dimension: (0.6, 0.8, 0.9)
  orientation: (0, 3.14, 0)
```

Also a television cannot be left out from the room is it already too dull. We place the television in the room facing the chair.

```
Television:
  position: (1.0, 2.0, 0.7)
  dimension: (0.8, 0.2, 0.45)
  orientation: (0, -3,0, 0)
```

A television is an interesting object. The German word (fernseher) can be literally translated as 'seeing far'. The television is an object that can make us perceive a virtual world by vision. Usually a film displays it's own perspective of a virtual world with little respect to the physical space it is displayed in. This is simply not needed for film. Suppose we would place a second television in the room and we would want it to display the same scene from the first television. We want it to respect the perspective the spectator is in. Then the position of the first television, the second television, and the spectators needs to be known. If we need those positions we might as well make sure these positions match the physical space. Matching the physical and the virtual realm's coordinates is an essential ingredient for mixing multiple media in multiple realities. We refer to these developments as Mixed Reality.

Since we now have a television which can act as a window to our virtual world we can add our first virtual object. We will add a simple sphere which is hovering in the middle of the room.

```
Ball:
  position: (1.0, 2.0, 1.0)
  dimension: (0.2, 0.2, 0.2)
  orientation: (0, 0, 0)
```

Suppose we would let the ball bounce through the room indefinitely. If this was a physical ball we would be able to see it. However since it is a virtual ball the only way to see it would be through the television. The only moments we would be able to witness the ball through the television would be when the ball would be within the vision line of you (the spectator) and the television. We already know the position of the ball. We know the position of the television. We only don't know the position of you. In this scenario we will register the position of your eyes.

```
Spectator:
  position: (1.0, 1.0, 1.6)
  dimension: (0.6, 0.3, 1.7)
  orientation: (0, 0, 0)
```

We can now have enough information to render the ball and display it on the television. A virtual camera will be placed at the exact position of your eyes. It's orientation will be perpendicular facing the television and its lens will be shifted so that it exactly matches the frame of the television. You can now see the ball through the television when it moves through the room.

5.1. Overview of data:

```
root:
WGS84: (52.1153016, 5.1389998)
Room:
  position: (0,0,0)
  dimension: (2.0, 4.0, 2.0)
  orientation: (radians? Euler?)
  matrix: (describes, position, rotation and
           scale in 9 floats)
```

```
Chair:
  position: (1,5, 3.5, 0)
  dimension: (0.6, 0.8, 0.9)
  orientation: (0, 3.14, 0)
Television:
  position: (1.0, 2.0, 0.7)
  dimension: (0.8, 0.2, 0.45)
  orientation: (0, -3,0, 0)
Ball:
  position: (1.0, 2.0, 1.0)
  dimension: (0.2, 0.2, 0.2)
  orientation: (0, 0, 0)
Spectator:
  position: (1.0, 1.0, 1.6)
  dimension: (0.6, 0.3, 1.7)
  orientation: (0, 0, 0)
```

6. Acknowledgments

We would like to thank the z25.org Foundation and MAPLAB of the HKU University of the Arts for supporting this research. As this research is referring to very common systems and facts we are referring to Wikipedia pages as they are more accessible in-depth documents than many scientific documents.

7. References

- [1] A. Loonstra, "Orchestrating computer systems, a research into a new protocol," *FOSDEM 2015 conference, Brussels*, 01-Feb-2015. [Online]. Available: <https://fosdem.org/2015/schedule/event/deviot02/>. [Accessed: 08-Apr-2015].
- [2] "Metric system," *Wikipedia, the free encyclopedia*. 15-Jul-2015.
- [3] "Rotation formalisms in three dimensions," *Wikipedia, the free encyclopedia*. 07-Jul-2015.
- [4] "World Geodetic System," *Wikipedia, the free encyclopedia*. 15-Jun-2015.
- [5] J. Burbank, \diamond D. M., and \diamond W. K., "Network Time Protocol Version 4: Protocol and Algorithms Specification." [Online]. Available: <https://tools.ietf.org/html/rfc5905>. [Accessed: 15-Jul-2015].