gyroscope

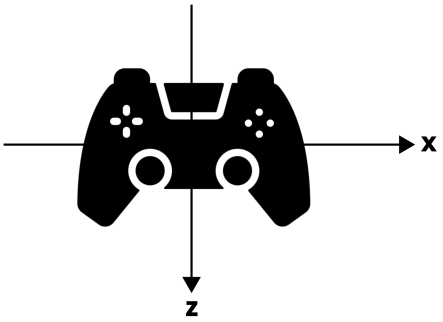
Albert Gräf <aggraef@gmail.com>, 2024-11-18

Some game controllers and many smartphones have sensors for detecting orientation and angular movement in 3-dimensional space, so-called inertial measurement units, better known as *gyroscopes*. joyosc (when invoked with the `-s` option) and TouchOSC (with scripting) can read the motion data of a device and convert it to OSC data. In joyosc, motion data is converted to `accel` and `gyro` data, each with 3 values (x, y, z). The accelerometer data (`accel`) measures acceleration in m/sec^2. This normally includes an acceleration counteracting gravity at 9.81 m/sec^2, so that it is possible to determine the orientation of the device relative to the ground. The angular movement data (`gyro`) measures rotation in radians/sec.

In order to correctly interpret this data, it is important to know the coordinate systems being used, which is where gamepads and smartphones differ.

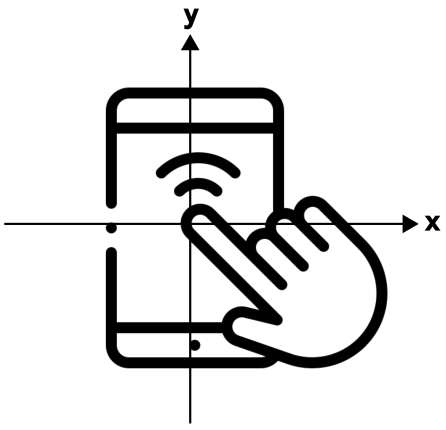
Gamepads

The principal plane of the device is the x-z plane, with the *z axis* pointing towards you, and the *y axis* pointing upwards, so if the device lies flat on a table, the *y acceleration* value will be positive, matching the value of gravity (at 9.81 m/sec^2). Note that if you hold the device upright, the *z acceleration* value will be *negative* (at -9.81 m/sec^2), as the z axis will then point downwards. In contrast, if the device is upright, but with its *left* side pointing towards the ground, the *x acceleration* value will be *positive* (at 9.81 m/sec^2).



Smartphones

On modern smartphones, the principal plane of the device typically is the x-y plane, with the *y axis* pointing away from you, and the *z axis* pointing upwards, so if the device lies flat on a table, the *z acceleration* value will be positive, matching the value of gravity (at 9.81 m/sec^2). If the device is upright, the *y acceleration* value, in contrast to the case of a gamepad, will be *positive* (at 9.81 m/sec^2). If the device is upright, with its *left* side pointing towards the ground, the *x acceleration* value will also be positive (at 9.81 m/sec^2).



Notes

- The somewhat quirky axis configuration of gamepads (with the y axis pointing upwards and the z axis pointing towards the player) apparently had its origin in the PlayStation controllers. This set the standard and is employed in games where gyroscope data is being used for aiming or moving the gamer's view. See the Gyro (AKA Motion Controls) Beginners Guide on YouTube for an introduction to the use of gyroscopes in 3D computer games.
- Another point worth mentioning is that the so-called *gravity vector* with a magnitude of 9.81 m/sec^2 included in the accelerometer readings is pointing *upwards* (away from the Earth's center). At first this might seem counter-intuitive, but the rationale here is that the device needs to be accelerated *away* from the Earth's center so that it doesn't fall to the ground. So there is a force being applied to the object counteracting the force of gravity, and this is what is included in the accelerometer data.
- Whether in Pd or in computer games, raw gyroscope data is hard to use, and its quality, precision, and update rate depend on the device at hand. Therefore, for most applications you will have to process the data in some way, in particular, to smooth it out and to determine the device's gravity vector and thereby its orientation in 3D space. Fortunately, there are algorithms for that which have been developed for use in computer games. See the gyrowiki by Jibb Smart (of Fortnite fame) for details.
- While most smartphones have gyroscopes these days, many game controllers don't, or may only provide the data to certain host devices. Xbox-compatible controllers usually come without gyros, because Microsoft's game controller APIs don't support that kind of data. However, the SDL library, which is being used by joyosc, TouchOSC, and many other programs, does support motion data. Controllers generally have gyroscopes if they support the Nintendo Switch or the SONY PlayStation gaming consoles. Examples of these are the the Nintendo Switch Pro and the PlayStation DualShock and DualSense controllers, for which many 3rd party clones are available. Other examples at the time of this writing are the 8bitdo Pro 2 and Ultimate Bluetooth & 2.4g, the GameSir Nova, Cyclone, and Cyclone Pro, and the Gulikit KK3 controllers.

Software

The toplevel directory contains some software for testing purposes:

- The controller-test.pd patch provides a simple way to show the raw gyroscope data produced by `joyosc -s`, along with button presses, joystick axes, and touchpad data (if available).
- A much more advanced version of this patch is available in gpmotion-help.pd. This lets you show preprocessed gyroscope data using Jibb Smart's algorithm mentioned above which calculates gravity and accumulated or sampled rotation data from the raw gyroscope inputs. Please check the gyrowiki for details. **NOTE:** This patch requires the gpmotion external which first needs to be compiled before the patch will work. Just running `make` will usually do the trick; see the toplevel README file for details.
- The TouchOSC (mk2) template joyosc.tosc can be used on Android and iOS devices to emulate a game controller including gyroscope data from the device (turn on the corresponding toggle in the template). It produces OSC data compatible with that of the joyosc program, so it can be used with the controller-test.pd patch in lieu of a real gamepad.