OMI_seat

Contributors

- Aaron Franke, The Mirror Megaverse Inc.

Status

Open Metaverse Interoperability Group Stage 2 Proposal

Dependencies

Written against the glTF 2.0 spec.

Optionally depends on the `OMI_physics_body` spec for triggers.

Overview

This extension allows defining a glTF node as a seat which a humanoid character can sit on.

A seat is defined by specifying the back, foot, and knee positions on the seat from which can be derived the IK positions and directions for placing a humanoid character on that seat. Optionally, the angle between the upper leg and spine can also be defined. This definition supports seats placed in any orientation and supports characters of any size sitting on them.

In addition to being defined independently, the `OMI_seat` extension can be defined on an `OMI_physics_body` node with the type set to trigger. When `OMI_seat` is on a trigger, the trigger should be treated as an entry for the seat, such that a player activating that trigger should cause that player to enter the seat.

Alternate seated positions, such as a tiny character with straight legs on top of the seat, or a character with their legs crossed on top of the seat, are not the recommended use cases of the seat definition, but are not forbidden if an app wants a character to sit in an unusual position. The primary purpose of this spec is to define the seat itself; how characters sit on that seat is secondary, and the fine details are up to the implementation.

This extension does not define any animations for entering or exiting the seat. It also does not define anything that a character's arms, head, tail, or chest should be doing, it only defines information about the seat itself intended to be used to place the legs on the seat.

Example:

This example defines a seat using a simple chair facing positive Z. This is the same as the `simple_chair_pos_z.gltf` example, but without the mesh data.

```
{
    "asset": {
        "version": "2.0"
    },
    "extensionsUsed": [
        "OMI_seat"
    ],
    "nodes": [
        {
            "mesh": 0,
            "name": "SimpleChairMesh"
        },
        {
            "children": [
                0
            ],
            "name": "SimpleChairPosZ",
            "extensions": {
                "OMI_seat": {
                    "back": [0, 0.33, -0.24],
                    "foot": [0, -0.25, 0.3],
                    "knee": [0, 0.33, 0.3]
                }
            }
        }
    ],
    "scene": 0,
    "scenes": [{ "nodes": [1] }],
}
```

More example assets can be found in the examples/ folder.

glTF Schema Updates

This extension consists of one new data structure for defining the seat limits, `"OMI_seat"`. The key `"OMI_seat"` can be added to the node-level `"extensions"` of a glTF node to define a seat.

The extension must also be added to the glTF's `extensionsUsed` array and because it is optional, it does not need to be added to the `extensionsRequired` array.

Property Summary

	Type	Description	Default value
back	`number[3]`	The position limit on the seat corresponding to the character's back in local space.	Required, no default
foot	`number[3]`	The position limit on the seat corresponding to the character's feet in local space.	Required, no default
knee	`number[3]`	The base position on the seat corresponding to the character's knees in local space.	Required, no default
angle	`number`	The angle between the spine and back-knee line in radians.	1.570796… (τ/4 or π/2 rad, 90 deg)

To avoid confusion with different standards of which direction is the forward direction, we define the forward direction by the direction from the back position to the knee position. The up direction is a vector perpendicular to the forward direction such that the direction from the knee to the foot points down. The back, foot, and knee positions should not be colinear or nearly colinear.

Back

The `"back"` property is a vector as a three-number array that defines the position limit of the character's back and hips in local space.

The character's upper leg (thigh) bones are typically placed on lines parallel to the line between the knee and back points, with the lines offset upward based on the thickness of the upper leg and offset to the side based on the desired distance between each leg. If the upper leg is smaller than this line, then the knees are aligned with the knee point, and the hips are moved forward in-between the knee and back points. If the upper leg is larger than this line, then the hips are aligned with the back point, the knees are moved forward farther than the knee point. The knees may also be moved up depending on the foot-knee constraint.

Foot

The `"foot"` property is a vector as a three-number array that defines the position limit of the character's feet and lower legs in local space.

The character's lower leg (calf) bones are typically placed on lines parallel to the line between the knee and foot points, with the lines offset forward based on the thickness of the lower leg and offset to the side based on the desired distance between each leg. If the lower leg is smaller than this line, then the knees are aligned with the knee point, and the feet are moved upward in-between the knee and back points. If the lower leg is larger than this line, then the feet are aligned with the foot point, and the knee is moved upward farther than the knee point. The knees may also be moved forward depending on the back-knee constraint.

Knee

The `"knee"` property is a vector as a three-number array that defines the base position of the character's knees in local space.

Angle

The `"angle"` property is a number that defines the angle between the spine and the line connecting the back and knee positions. If not specified, the default value is `1.57079632679489662` or τ/4 or π/2, which is 90 degrees represented in radians.

This is an angle instead of a position because:

- We want to support cases where the character in the seat is smaller than the seat, in which case the hips may be far from the back position in order to place the knees at the knee vector, and the angle between the hips and an upper position would change. Instead we just want to define which direction the spine points.
- We do not want to define the upper bound of a seat. A seat should support arbitrarily large characters (if an app wants to limit character size, that's an application-specific constraint beyond the scope of seats), and there is no way to IK a large character within such a constraint without breaking their back or violating other constraints, unlike with the foot/knee/back position.
- We want to make this optional for simple use cases. Defining an angle between the spine and back-knee line is unnecessary information for most office chairs, dinner table chairs, and seats without a solid back.

This property only needs to be defined when the desired angle between the spine and back-knee line is not 90 degrees. For example, a reclining chair that has a back tilted so the character could lie back should have this defined.

Property Details

The positions defined above can be used to uniquely find other vectors used to place a character on the seat.

- The upper leg main direction is the direction from the back vector to the knee vector. This is also the forward direction.
- The lower leg main direction is the direction from the knee vector to the foot vector.
- The right direction is the normalized cross product of the lower leg main direction and the upper leg main direction.
- The upper leg up direction is the cross product of the right direction and the upper leg main direction.
- The lower leg forward direction is the cross product of the right direction and the lower leg main direction.
- The spine direction can be found by rotating the upper leg main direction around the right vector by the angle property.

These are the directions of the seat itself, the actual directions a character ends up at may be different. The specifics of this depend on the IK system used. However, one common feature is the need to offset from the seat based on the thickness of the legs in order to determine the points to use for IK, so that we avoid the character's legs being inside of the seat itself.

- The IK position for the back/hips/spine is the back position, plus the upper leg thickness times the upper leg up direction, plus the spine thickness times the upper leg main direction.
- The IK position for the feet is the foot position, plus the lower leg thickness times the lower leg forward direction, minus the foot thickness times the lower leg main direction.
- The IK position for the knees is more complicated. It needs to be adjusted by both the upper leg and lower leg offsets. For perpendicular upper and lower legs, this is trivial, you just add the vectors together. Otherwise some trigonometry is required. The knee offset can be found by adding the upper leg offset with the upper leg direction multiplied by an adjustment scalar. The adjustment scalar can be found with the lower offset length divided by the sin of the angle between the legs, minus the upper offset length divided by the tan of the angle between the legs.

In a nutshell, the legs sit as close to the seat as possible while not clipping though it, therefore we offset by the leg thickness, otherwise the leg bones would be exactly on the seat's surface. We want to keep the seat's points exactly on the seat and have the character legs offset from the seat's points so that we don't play favorites for any particular leg size.

JSON Schema

See schema/node.OMI_seat.schema.json for the schema.

```
{
    "$schema": "http://json-schema.org/draft-04/schema",
    "title": "OMI_seat glTF Node Extension",
    "type": "object",
    "required": [
        "back",
        "foot",
        "knee"
    ],
    "properties": {
        "back": {
            "type": "array",
            "description": "Three number array defining the seat position corresponding to the character's ba
        },
        "foot": {
            "type": "array",
            "description": "Three number array defining the seat position corresponding to the character's fe
        },
        "knee": {
            "type": "array",
            "description": "Three number array defining the seat position corresponding to the character's kn
        },
        "angle": {
            "type": "number",
            "description": "The angle between the spine and back-knee line in radians.",
            "default": 1.57079632679489662
        }
    }
}
```

Known Implementations

- Godot Engine add-on

Resources:

- None