

X3D Types (attributes)

SFField (Python tuples immutable)
 FBool bool

SFInt32 int
 SFFloat float (exact precision)
 SFDouble float
 SFString str or String (or both?)
 SFVec2F 2-tuple float: (float, float)
 SFRotation (float, float, float, float)

MFField (Python lists are mutable)

MFBool [bool]
 MFInt32 [int]
 MFVec2F [(float, float)] ↔ [SFVec2F]
 MFString [str] ↔ [String]

SFNode / MFNode: deferred

X3D Elements Python classes

Node, Structures
 Simple fields have explicit default values as class constants
 Node fields are ...

-init-
 # set all properties (no others allowed)
 # per 3-way keyword call worker

property handling using class types

-validate- # potentially unnecessary?
 # intrinsic to correct design

RTE messages, identify node, field, reason
 # include bounding annotations

Examples

```
<Group DEF="A" when True = "1 2 3" />
X3D.Group("A")
node A = X3D.Group(1; nodeA.DEF="A"; nodeA.when True=(1 2 3)
node A.children = node B # once node exists
# child A = node B
# child B = X3D.Group(1)
X3D.Group(DEF="A", when True=(1 2 3), children=X3D.Group(1))
```

Design Goals

Let Python be Python... for V3D architecture.
 Class abstractions, @ut files
 Auto-generate best-practice patterns via X3DOM
 Flat namespace, no need for programmer to know a node's component
 package utilization
 from X3D import *

prefix defaults but not attributes X3D.Transforming (---)
 lock-down attributes, no inadvertent additions (via python properties)
 valid scenegraph structure & values throughout. loading invalid scene → error-specific RTE.
 Not a goal: writing an X3D renderer, have emphasis on data structures vica "performance"
 Support Big Data pipeline presentation outputs, eventually

Issues

Zero/Minimal dependencies: numpy not needed but separately installable
 Injurious binding: import X3D as X3D thus similar/identical program syntax
 Similar to python module/join, load/eval? Maybe... but need state main serializations (inputOnly/outputOnly)
 Goal: defining all node values in single line as an optional. TBO: inputOnly, outputOnly transient methods
 lazy initialization of property values as needed: get correct property patterns
 - if unused X3D nodes (over 255 total) using correct package/cls pattern
 Providing X3D abstract node types: internal only. X3DOM easily could add or modify Python ABC classes. TBO.

